MULTI-VIEWPOINT LANE DETECTION WITH APPLICATIONS IN DRIVER SAFETY SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Amol Borkar

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2012

MULTI-VIEWPOINT LANE DETECTION WITH APPLICATIONS IN DRIVER SAFETY SYSTEMS

Approved by:

Dr. Monson H. Hayes III, Advisor
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Magnus Egerstedt
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark T. Smith, Co-Advisor
Professor, School of Information and Communication Technology
Swedish Royal Institute of Technology (KTH)

Dr. Nikil Jayant
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David V. Anderson
Associate Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Brani Vidakovic
Professor, Deptartment of Biomedical Engineering
Georgia Institute of Technology

Date Approved: 9 December, 2011

This dissertation is dedicated to my family,

Anil, Gauri and Milind Borkar:

Without you, I would not be where I am today.

ACKNOWLEDGEMENTS

During the course of my doctorate at Georgia Tech, I have met a number of people that influenced many of my decisions. These decisions put me on the right path to develop a novel scientific thesis and ultimately brought me to where I am today. As it will be practically impossible for me to thank each person that has had an effect on my life; I will, nonetheless, acknowledge those who have made a substantial impact on me as a person, and as a researcher.

I would like start by thanking my family. My father, Anil Borkar, taught me to excel in everything I did. He also provided me with all the necessary resources to help shape my academic career. My mother, Gauri Borkar, made my life her priority and put her own career behind. She taught me to fight for what I believed in, and to never give up. Lastly, my brother, Milind Borkar, was a great mentor whose work inspired me to take up the Ph.D. at Georgia Tech.

I would like to thank my first Ph.D. advisor, Dr. Oskar Skrinjar. Working with him and his group on medical imaging projects set my interests in computer vision and image processing. Regrettably, after one year of substantial progress, I had to leave his great team. Fortunately, I was introduced to my current advisor, Dr. Monson Hayes. He is very well rounded and has an arsenal of excellent ideas. Dr. Hayes has also been a great friend whose company I have enjoyed outside of academia. I cannot forget my co-advisor, Dr. Mark Smith, whom I met while working on a face recognition personalization project. His always positive attitude and support helped me continually improve my work. I honestly could not ask for a better team of advisors. Dr. Hayes provided the guidance from an academic point of view, while Dr. Smith provided the industrial perspective to the research. Together, they did

an incredible job of pointing me in the right direction and helped shape what has become my thesis.

Dr. David Anderson is a member of my defense committee that I have known for many years. He is a creative thinker and is very easy to approach. He has provided great guidance and suggestions that helped improve my thesis. I would also like to thank the remaining members of my defense committee: Dr. Magnus Egerstedt, Dr. Scott Wills (formerly on the committee), Dr. Nikil Jayant, and Dr. Brani Vidakovic, who took time out of their busy schedules to be a part of this monumental milestone in my life.

Being a member of CSIP, I have had the opportunity to interact with some the most intelligent people from all over the world. During the rut times in research, these people have provided valuable inputs that in various ways helped improve the existing research: Mohammed Aabed, Vikram Appia, Salman Asif, Salman Aslam, Guillermo Bonnet, Volkan Cevher, Nicolas Gastaud, Balaji Ganapathy, Musad Haque, Aditya Joshi, Kaustubh Kalgaonkar, Jinwoo Kang, Irteza Khan, Sourabh Khire, Greg Krudysz, Yeongseon Lee, Brett Matthews, Aleem Mushtaq, Devangi Parikh, Hrishikesh Rao, Michael Santoro, Mashour Solh, Smita Vemulapalli, Emily Xu. Aside from the CSIP members, I would also to thank Vishal Patel (University of Maryland, College Park), a friend of mine from my undergraduate studies with whom I have always had interesting technical discussions on computer vision research.

Lastly, I would like to thank the ECE administrative staff, namely Cordai Farrar, Lisa Gardner, Catherine Gholson, Marilou Mycko, Tammy Scott, Stacie Speights, Tasha Torrence, Jacqueline Trappier, and Suzette Willingham who always made sure my administrative needs were always handled smoothly and efficiently.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this dissertation is to develop a Multi-Camera Lane Departure Warning (MCLDW) system and a framework to evaluate it. A Lane Departure Warning (LDW) system is a safety feature that is included in a few luxury automobiles. Using a single camera, it performs the task of informing the driver if a lane change is imminent. The core component of an LDW system is a lane detector, whose objective is to find lane markers on the road. Therefore, we start this dissertation by explaining the requirements of an ideal lane detector, and then present several algorithmic implementations that meet these requirements. After selecting the best implementation, we present the MCLDW methodology. Using a multi-camera setup, MCLDW system combines the detected lane marker information from each camera's view to estimate the immediate distance between the vehicle and the lane marker, and signals a warning if this distance is under a certain threshold. Next, we introduce a procedure to create ground truth and a database of videos which serve as the framework for evaluation. Ground truth is created using an efficient procedure called Time-Slicing that allows the user to quickly annotate the true locations of the lane markers in each frame of the videos. Subsequently, we describe the details of a database of driving videos that has been put together to help establish a benchmark for evaluating existing lane detectors and LDW systems. Finally, we conclude the dissertation by citing the contributions of the research and discussing the avenues for future work.

CHAPTER I

INTRODUCTION

1.1 Overview

The first vehicle was invented in the 18th century as a means of transporting passengers or goods from one place to another. Through the centuries, innovations in a number of technical disciplines have helped in improving nearly all aspect of its initial design. Consider the speed of travel. By allowing the vehicle to reach high speeds, the duration of the commute can be reduced. However, an increase in speed tends to have a negative impact on driver safety as the driver has less time to react to an impending situation. For example, on freeways and highways where the vehicles tend to commute at high speeds, the consequences of a driver error or distraction can be fatal. Over the years, high fatality rates relating to traffic accidents have been reported across the globe. In Southeast Asia, China's Ministry of Public Safety reported 667,507 deaths related to traffic accidents in 2003 [8] while Malaysia reported a total of 35,425 causalities in 2006 [9]. That same year, Indonesia and Vietnam reported 16,548 and 12,800 casualties respectively [9]. In the Middle East, countries such as Iran, Egypt, Saudi Arabia reported 22,918, 12,295, and 6,358 fatalities respectively in 2007 [9]. Among the countries in the Americas, the US has the highest statistic for traffic related deaths. To elaborate, Brazil, Canada, and Mexico reported motor vehicle related fatalities at 35,155, 2,889, and 17,003 respectively in 2006 [9]. That same year, the US reported a total of 42,642 casualties [9]. Furthermore, in 2009, the Center for Disease Control and Prevention (CDC) in the US reported a total of 2.43 million fatalities [10] of which 34,000 were traffic related [11]. Finally, in a study performed by Jacobs et al. [12], the total number of motor vehicle related deaths

around the world was estimated at 800,000 in 1999. This number is expected to increase to 1.3 million by 2020 [9, 12].

Observing the statistics listed above, it is evident that traffic accidents account for a large number of fatalities world-wide. Consequently, exploring new avenues that improve driver safety are of interest to the automotive industry. One of these avenues is through the use of Driver Assistance systems.

1.2 Driver Assistance Systems

Driver Assistance (DA) systems are systems that provide aid or assistance to the driver while driving. These systems operate by taking input from sensors around the vehicle to compute some form of feedback that is then used to assist the driver of the vehicle. An example of a DA system is Adaptive Cruise Control (ACC). ACC uses a radar to monitor and keep a pre-defined distance to the vehicle ahead, thereby augmenting the existing cruise control system found in most vehicles [1]. ACC is illustrated in Fig. 1 where the blue car is able to maintain a safe distance to the silver car ahead by analyzing the radar signal (shown by the blue beam). Besides radar,

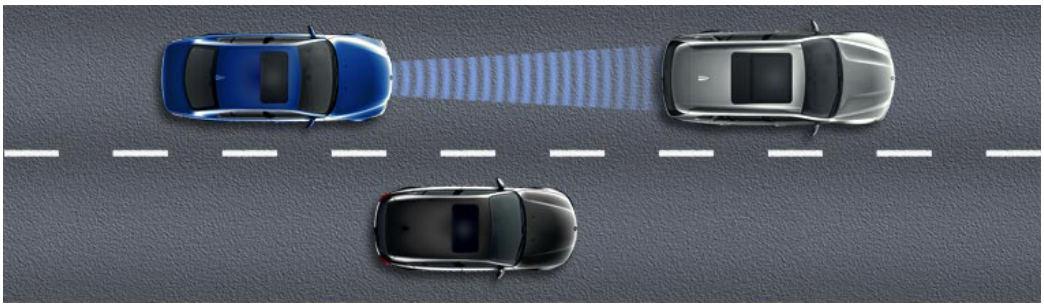


Figure 1: Adaptive Cruise Control [1].

GPS, camera, LIDAR and other sensors have also been used to create a variety of DA systems. However, this document focuses only on camera based systems, specifically lane detection and Lane Departure Warning (LDW) systems.

1.3 Camera based DA systems

Camera or vision based lane detection can be described as a problem of estimating the lane boundaries by locating white or yellow markers on the road surface with little knowledge of the road geometry. The hardware for a lane detection system consists of single or multiple cameras that look out of the front windshield and have a view of the road ahead. These cameras acquire data in the form of images of the road surface that are analyzed to extract features that correspond to the desired lane markers. Lane detector hardware and lane markers as they appear on the road are shown in Fig. 2. Lane detection will be described in detail in Chapter 2. Lane detection is of

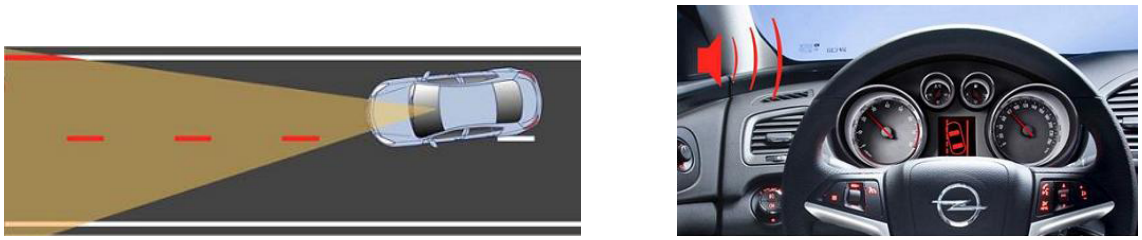
(a) Lane detector hardware [13].

(b) Lane markers on the road.

Figure 2: Lane Detection [2].

much interest to automotive research because it serves as the foundation for many DA systems such as Lane Bobbing Detection (LBD), Lane Departure Warning (LDW), and Blind Spot Monitoring (BSM) to name a few. Lane Departure Warning is also an interesting DA system that constantly monitors the position of the vehicle with respect to the lane markers on either side. If the vehicle comes within a certain distance of a marker, the driver is notified and a corrective measure can be undertaken. Using LDW, unintentional lane departure caused by driver distractions, fatigue, or driving under the influence of a controlled substance can be reduced [14]. An illustration of

an operating LDW system is shown in Fig. 3 where a vehicle is about to change lanes and an alarm is sounded in the cabin [2]. LDW and its involvement in the MCLDW system will be discussed in detail in Chapter 3.



(a) Vehicle crosses over lane boundary. (b) Alarm sounded inside the cabin.

Figure 3: Lane Departure Warning [2].

Besides the technology that drives these DA systems, a framework to evaluate DA systems is also presented in this dissertation.

1.4 Evaluation Framework

The evaluation framework consists of two components: a procedure to create the ground truth, and a database of videos. The creation of ground truth is performed by a technique called Time-Slicing. Time-Slicing is a quick and efficient way to speed up the process of annotating the true locations of the lane markers in a sequence of images [15]. The availability of the ground truth enables the performance of both the lane detection, and the LDW systems to be objectively quantified. Time-Slicing and its advantages will be covered in Chapter 4. Chapter 5 then goes into the details of the database of driving videos. The database is made up of numerous image sequences that are recorded during various times of the day to provide a realistic set of conditions that one encounters while driving. This also prevents a bias in the performance of a DA system that could be tailored to operate well only in specific environmental conditions.

1.5 Organization of the Dissertation

This dissertation focuses on the development of a novel DA system that we call the Multi-Camera Lane Departure Warning (MCLDW) system. In addition, we also present the framework to evaluate the MCLDW system. The dissertation is organized as follows:

Chapter 2 focuses on the development of the lane detection component of the MCLDW system. In this chapter, three different lane detection methodologies are implemented. Each implementation undergoes preliminary testing and their results are presented. Additionally, a competing algorithm is also implemented. Then, all the implementations are compared and their performances are quantified objectively. Finally, the best performing implementation is selected for use in the MCLDW system.

Chapter 3 introduces Lane Departure Warning (LDW) and presents the MCLDW methodology. This chapter also describes the details of camera calibration and the procedure used to estimate the immediate distance to the lane markers on either side of the vehicle. Finally, the MCLDW system is compared to a single camera LDW system and the performances of the two systems are objectively quantified.

Chapter 4 is the first of two chapters that cover the evaluation framework. It focuses on explaining the procedure for creating ground truth using the Time-Slicing approach. Following the introduction, a detailed explanation on annotating the true locations of the lane markers in a video clip is provided. Then, the ground truth created using the Time-Slicing and the entirely manual approach are compared. The chapter is concluded by discussing the advantages of using the Time-Slicing approach.

Chapter 5 presents the database of videos that has been created for the purpose of evaluating the lane detection and the LDW systems. Details regarding the acquisition hardware, the data formats, and the pipeline for quality control that are required in the composition of the database are discussed. Finally, access information to the database is provided.

Chapter 6 lists the publications that have resulted from this research. In addition, the contributions of the publications that formulated the basis of the dissertation are also provided.

Chapter 7 presents the summary and conclusion reported in this dissertation. In addition, avenues to continue and extend the presented research are also discussed.

The dissertation is designed to be self contained and the reader is expected to have basic understanding of image processing and computer vision techniques.

CHAPTER II

LANE DETECTION

2.1 Overview

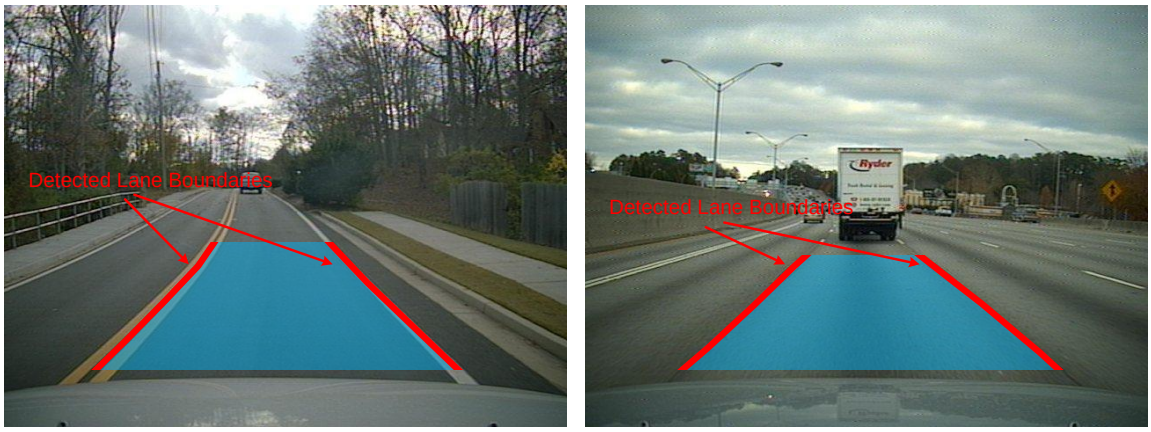
As discussed in the introduction, Driver Assistance (DA) systems are systems that provide aid or assistance to the driver while driving. DA systems function by taking input from a variety of sensors around the vehicle to process feedback that is then used to assist the driver of the vehicle. A perfect example of a DA system is Adaptive Cruise Control (ACC). Here, a radar sensor is used to determine the distance to the vehicle ahead. This distance is then used as an input by the ACC system to adjust the speed of the host vehicle during a cruise, while also maintaining a pre-defined minimum distance to the vehicle ahead [1]. Besides radar, examples of other sensors that have been used in DA system include LIDAR/LADAR (Light Detection And Ranging), GPS (Global Positioning System) and cameras to name a few. However, the focus of this chapter is on camera based DA systems. This type of DA system often consists of one or more cameras that are placed in a manner such that they have a view out of the windshield. The cameras acquire images that are processed to determine features based on the desired application. Some examples of camera based DA systems include LDW (Lane Departure Warning), LCW (Lane Change Warning), and Intelligent Headlight Control (IHC). In LDW, the position of the vehicle is constantly monitored within a lane. If the vehicle is within a certain distance of a lane boundary, then the driver is notified of lane departure unless certain criteria are met [16]. Similarly, in LCW, the blind spot of the vehicle is constantly monitored for oncoming traffic. If a vehicle is detected in the neighboring lane, then the driver is notified that it is unsafe to change lanes [17]. Finally, in IHC, the use of the high

beam of the headlight is maximized at night. This is done by switching the headlight from high beam to low beam only in the presence of neighboring vehicles [18]. A component that is common to each of the examples described above is lane detection. For example, in LDW, the position of the vehicle within the lane can be determined by detecting the position of the lane boundaries with respect to the vehicle. Similarly, in LCA, lane detection is used to determine if the detected traffic is in the same lane as the host vehicle or in the neighboring lanes. Lastly, in IHC, lane detection is used to determine if the detected headlights or taillights belong to vehicles on the road and not artifacts, thereby reducing false detections. Based on this discussion, it is clear that lane detection is an important topic. Therefore, the purpose of this chapter is to cover lane detection in detail: how it is done, how it is tested, how it is evaluated etc. But, before we move ahead with more technical details, let us first establish some definitions.

A **Lane** is defined as part of a paved road that is marked out for use by a single file of vehicles travelling in the same direction. The boundaries of a lane are designated by the lane markers and are represented as a curve. Lane markers come in two varieties: solid, and dashed as shown in Fig. 4. Details regarding the specifications

Figure 4: Solid and dashed lane markers in an image.

and applications of each lane marker can be found by referring to the Federal High Adminstration's (FHA) handbook [19]. **Lane Detection** is defined as a problem of estimating the left and right boundaries of a lane that are designated by the solid or the dashed markers. For a multi-lane road, a lane detector finds only the immediate left and right lane boundaries of the lane. In Fig. 5, we have shown a lane and its boundaries for both a single lane and a multi-lane road.



(a) Single lane road. (b) Multi-lane road.

Figure 5: A Lane and the detected lane boundaries in an image. A lane is shown as a blue overlay on the road, and the lane boundaries are shown in red.

Having defined lanes and introduced the problem of lane detection, in the next section we will discuss the necessary requirements for a functioning lane detector. Then, we will go over some of the related research that has been performed in this field. Next, we will introduce three new lane detection methodologies. Each methodology contains several original components that will be covered in detail in this chapter. We will also introduce a procedure to systematically evaluate the lane detectors and report their performance. In addition, we will implement a competing algorithm and compare its performance to our methodologies. Finally, in the conclusion, the chapter will be summarized and the best performing lane detector will be determined.

2.2 Operation and Evaluation Requirements of a Lane Detector

The requirements for a complete lane detector are:

1. The lane detector must be able to detect the lane markers on the road.
2. The lane detector must be able to operate in the following conditions:
   (a) In the presence of stationary or moving vehicles ahead and in neighboring lanes.
   (b) In the presence of solid or dashed lane markers.
   (c) On straight and curving roads (both single and multi-lane).
   (d) In the presence of road surface texture variations and deformities.
   (e) In the presence of shadows and dynamic illumination conditions.
3. Lane detectors have evolved significantly over the years; hence, we are only concerned with technologies that were introduced in 2006 or later years.
4. The lane detector must be tested on at least 40 seconds of video or 1200 images that were recorded while driving on city roads or highways and at speeds above of 30 mph. No synthetic data.
5. The output of the lane detector must be either a set of analytic curves or a set of points that represent the estimates of the lane boundary curves in the image. The curves must be at least 20 ft in length as shown in Fig. 6.
6. The output must be quantified and tabulated using objective evaluations with regard to a reference or the ground truth (ground truth is covered in detail in Chap. 4). Objective evaluations prevent relying on visual inspection that can introduce bias in the results. Furthermore, the evaluations must be performed with regard to the entire lane boundary curve, and not just a specific location

Figure 6: Curve of length 20 ft.

on the curve. Besides only the correct detection rates, results should also contain the rates for missed detections, false positives, and error.

It should be noted that lane markers on the road do not follow a mathematical formulation, rather, they are designed to be discerning to the human eye. Hence, to detect lane markers, most existing lane detectors are built on a set of rules that are based on the appearance of lane markers in the image. These rules are often unique and differ from one lane detector to another.

2.3 Related Work

In this section, it was our intention to dissect the lane detector that is considered to be the "standard" for comparison. We consider an algorithm to be the standard if it is a sophisticated technique that shows good results and is often used as the basis for comparison in many publications. However, in the lane detection community, most publications do not compare their work to others. The root of this problem is the lack of availability of the source code. Thus, comparing to another technique requires implementing the technique using the limited details provided in the publications. Besides missing information in the publication, implementing someone else's work can

be very tedious and time consuming; hence, it is avoided. However, this problem with implementation can be avoided if testing is performed on common data sets. But, there are no such standardized data sets. As a result, most publication test their algorithms on their own data sets which are proprietary and not accessible to the public. In addition, there appear to be no established or standardized performance measures that are used to systematically evaluate the lane detectors. This is also contributed to by the lack of ground truth in almost all data sets. As a result, most evaluations are made based purely on visual inspection, and can introduce bias in the results. Based on the above reasoning, it is clear that comparing to other techniques is very difficult. As a result, establishing a common basis or the standard for comparison is not possible at the moment.

One could say that closely examining the techniques from few of the publications that have been cited numerous times is a good indicator of the common technique used for comparison. However, it should be noted that lane detection is a system level research problem. As a result, a lane detector consists of several components or blocks in its layout that holistically produce the result of lane marker localization. Therefore, if a publication has been referred to many times, the reference often corresponds to a particular component in the lane detector and not the entire lane detector itself. For example, the GOLD algorithm by Bertozzi et al [20] has been cited over 500 times. However, most of the publications that refer to the GOLD algorithm are only citing a particular component in its layout called Inverse Perspective Mapping (this will be covered in detail later in this chapter), and not the entire lane detector. Therefore, it can be misleading to examine a publication based purely on the number of times it has been referenced. As a result, we decided to consider only the publications that met our requirements from the previous section. In addition, we did not look very deeply into commercial products such as the lane detectors developed by Mobil-eye and Iteris. The reason for this was because commercial products do not provide details regarding

their implementation, the data used in testing, and the performance measures used in evaluation. As a result, it is not only impossible to replicate them, but the quality of their estimates produced is also questionable. In performing the literature review, we sifted through over 500 publications on lane detection and related fields. However, we were only able to find only one publications that meet all the requirements from Sec. 2.2. Hence, we will only take a look at this publication in some detail.

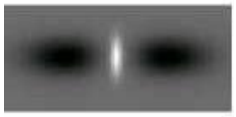
This algorithm was designed by Aly [3] and was used by Alice, Team Caltech's entry in the *2007 Darpa Urban Challenge* to find lane markers on the roads [21]. Its methodology begins by converting the camera captured image from color into a grayscale image that is composed only of the red color channel from the image. Then, using the camera's calibration parameters such as height, field of view angles, and focal length to name a few, the grayscale image is transformed into a bird's-eye view using the Inverse Perspective Mapping (IPM) transformation [20]. Details of this transformation will be covered in the following section. A sample input image and the transformed grayscale image are shown in Fig. 7. Then, the image is filtered using a 2-D kernel that is composed of the second derivative of Gaussian in the horizontal

(a) Camera image. (b) IPM transformed image.

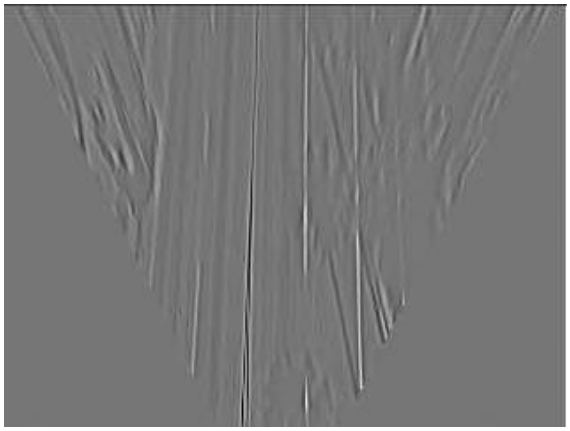
Figure 7: Transformation of the camera image to the IPM view [3].

direction, and a smoothing Gaussian in the vertical direction. Furthermore, this kernel is tuned to detect the locations of lane markers that are 6 inches wide. The

resultant filtered image is then subject to a threshold that is used to preserve the actual pixels values from the filtered image above the threshold without converting the image to binary. An example of the 2-D kernel, filtered image, and filtered image after the threshold are shown in Fig. 8. Then, using a line fitting algorithm called the Hough transform [22], and an iterative outlier elimination called Random Sample Consensus (RANSAC) [23], the lane markers are located in the filtered image. A



(a) Kernel used for filtering.



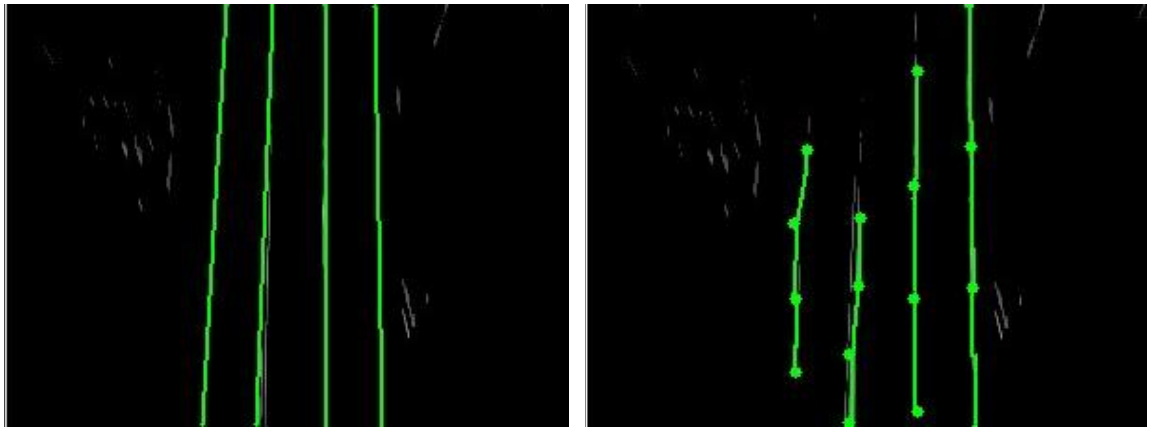
(b) Image after filtering.

(c) Image after thresholding.

Figure 8: Filtering the IPM view [3].

detailed explanation of both the Hough transform and RANSAC are provided later in this chapter. Fig. 9a shows the lines that are detected in the filtered image as a result of using the Hough transform and RANSAC. Next, a small window is set up around each detected line and RANSAC is performed again. The inliers that are detected by RANSAC are used in Bezier spline fitting. In each iteration of RANSAC, a new Bezier spline is generated and is assigned a score. This score is computed based on the curvature of the spline i.e. the scoring formula favors straighter splines producing a higher score over the shorter and curvier ones. The result of the spline

fit is a set of splines that represent the detected lane boundaries in the image. The detected splines in the filtered image are shown in Fig. 9b. Lastly, a post-processing

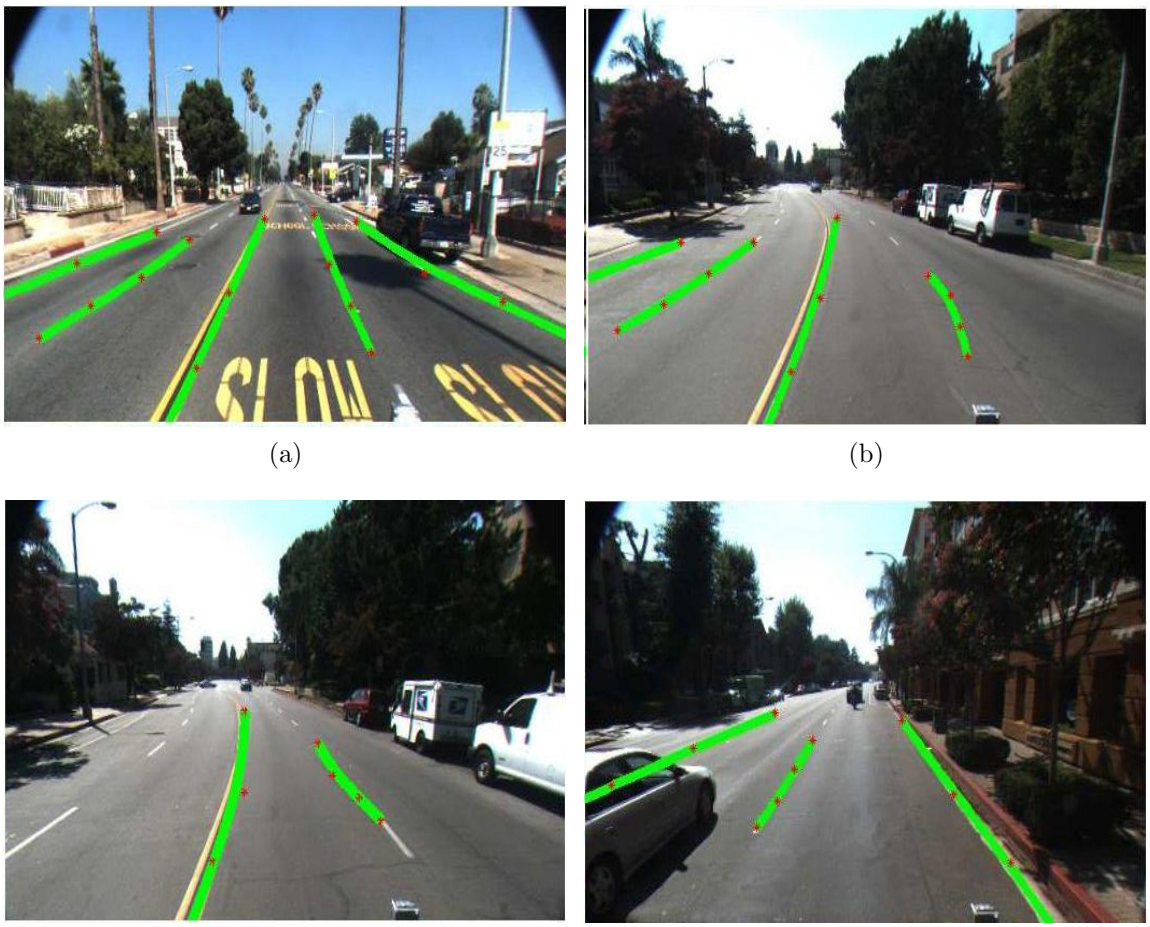


(a) Lane markers located in the filtering image. (b) Detected splines in the filtered image.

Figure 9: Lane marker localization and spline fitting [3].

step is used to try to better localize the spline and extend it in the grayscale image. The result of this methodology is a lane detector that is able to detect lane markers on city roads and highways, and in various illumination conditions. Some images of the estimated lane boundaries are shown in Fig. 10. The lane detector is tested on over 1200 images and is systematically evaluated by comparing the estimates to the ground truth along the entire length of the curve.

Although the above described methodology shows good results, it is expected to face a few problems. For example, the current approach applies a threshold and selects the pixels corresponding to the highest 2.5% of intensity values in the filtered image. This approach is expected to work well when objects corresponding to the 2-D kernel i.e. lane markers are present in the image. However, there may be times when lane markers may not be present on the road. In these situations, the filtered image will be populated with low intensity values corresponding to the artifacts and random objects on the road. However, these low intensity values may be among the highest 2.5% of intensity values in the filtered image. As result, the random objects and artifacts may cause false detection. Furthermore, RANSAC and spline fitting would use these false



(a) (b)

(c) (d)

Figure 10: Lane boundaries estimated while driving [3].

detections to produce estimates of the lane markers. Some examples of these incorrect estimates are shown in Fig. 11 where no lane marker is present but the algorithm accidentally estimates them on the road. In addition, the lane detector does not incorporate a tracker. Hence, the lane marker estimates are computed independently from one image to the next. Therefore, in the presence of noise or occasional false detections, the estimates of the lane markers may move and change their position rapidly between images. Finally, in Sec. III.A of the publication, it appears that the method used to compare the estimated lane boundary to the ground truth simply determines the pixel distances between the two curves. It does not account for the perspective effect in the image. As a result, this method provides more weighting to

(a) (b)

Figure 11: Incorrect lane boundaries estimates [3].

the errors that are near the vehicle in contrast to the errors that are near the horizon. However, the errors should have equal weighting regardless of their positions with respect to the vehicle.

Over the years, numerous approaches have been developed for lane detection, and each approach has produced varied results. The algorithm by Aly [3] is a very good approach at addressing the problem of lane detection. But, there are some situations that need fixing or improvement. Based on the discussions in this section, it is clear that lane detection is not a solved problem. This is the motivation for further research into new algorithms that produce better results. Therefore, in this chapter, we will introduce three original methodologies to perform lane detection. Each methodology is an improvement over the previous. We start with the first one in the next section.

2.4 *Advanced Lane Detector 1.0*

In this section, we will introduce the Advanced Lane Detector (ALD) 1.0, a new methodology that is used to perform lane detection. The layout of the lane detector is shown in Fig. 12. First, the camera captured images undergo Preprocessing to highlight the lane markers in the image. Next, in the Inverse Perspective Mapping (IPM) block, the preprocessed image undergoes a geometric transformation that removes the effect of perspective [20]. Then, in the Object of Interest Detection block, the transformed image is converted to binary using an adaptive threshold that helps detect lane markers. The Preliminary and Detailed Feature Extraction blocks per-

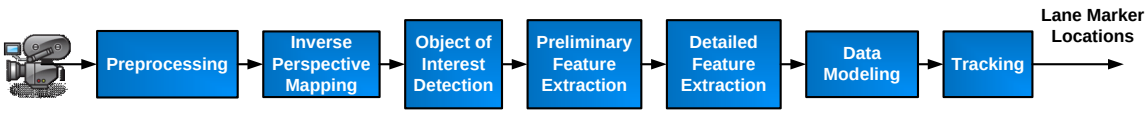


Figure 12: Layout of the ALD 1.0.

form a series of template matching procedures to find the lane markers in the image. Then, in the Data Modeling block, Random Sample Consensus (RANSAC) [23] is used to eliminate outliers while inliers that correspond mostly to lane markers are retained. Finally, a curve is fit onto the inliers to represent the estimates of the lane boundary curves, and then the parameters of the curve are tracked from one image to the next. The ALD 1.0 is tested with videos that reflect real world driving conditions and its performance is discussed later in the section. The details of all the blocks shown in the Fig. 12 are provided below.

2.4.1 Preprocessing

This is the first block of the ALD 1.0. The purpose of this block is to highlight the lane markers in the image making them more noticeable and easier to detect [24]. As shown in Fig. 13, the preprocessing block is further divided into two stages:

1. Temporal Blurring

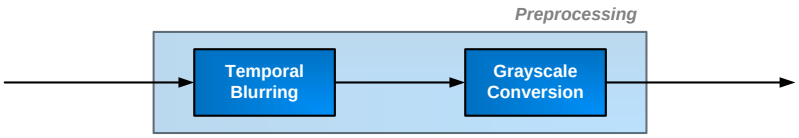


Figure 13: Stages in the preprocessing block.

2. Grayscale conversion

Earlier in this chapter, we mentioned that solid and dashed markers are the two types of lane markers that appear on the road surface. In general, the solid lane markers appear as long curves in an image; on the other hand, the dashed lane markers appear either as short segments or small dots in the image as shown in Fig. 14. As a result, the solid lane markers are often easy to detect while the dashed lane makers tend to be difficult to detect. Therefore, to aid in the detection of dashed

Figure 14: Solid and dashed lane markers in an image.

lane markers, Temporal Blurring is introduced. Temporal blurring creates an Average Image $\bar{I}_K(n)$ in which the dashed lane markers appear as a solid lane marker [24]. Consider a video clip that contains N images where each image is indexed by n and $n \in [1, N]$. The Average Image is created by computing the average between the

current image $I(n)$ and K images from the past. It is formed as follows:

$$\bar{I}_K(n) = \sum_{i=0}^{K} \frac{I(n - i \cdot \Delta)}{K} \tag{1}$$

where i and Δ are used to create the offset from the current image index n. Here, we used $K = 3$ and $\Delta = 3$ to create the Average Image. To determine the values for K and Δ, we perform quantitative reasoning. Let us consider a vehicle that is assumed to be travelling at a constant speed of 65 mph or 3.16 ft/33 ms where 33 ms is the time duration between each image for a 30 frames per second (fps) video clip. Based on the FHA's specifications, the dashed lane markers are 10 ft in length and have a gap of 30 ft between each marker [19]. Therefore, in the duration between 3 images or 99 ms, the vehicle will have travelled 10 ft or the lane markers have moved 10 ft closer to the vehicle from their initial position. Similarly, in the time between 6 and 9 images, the lane markers will have moved 20 ft and 30 ft, respectively. As a result, the lane markers in the three previous images ($I(n-3)$, $I(n-6)$, $I(n-9)$) will fall in the gap between the lane markers in the image $I(n)$. Therefore, the average of all four images shown in Fig. 15 produces an Average Image $\bar{I}_K(n)$ in which the dashed lane markers appear as a single, connected solid marker as shown in Fig. 16. At speeds above and below 65 mph, gaps may appear in $\bar{I}_K(n)$; however, these gaps will be much smaller than the gaps between the dashed markers in a single image. In Fig. 16, we see that the dashed lane markers on the right now have the appearance of a solid lane marker. Additionally, the solid lane marker on the left appears to be unaffected by the averaging.

In Fig. 17, we have shown four examples of Average Images. In Fig. 17a and 17b, the Average Images were computed while the vehicle is travelling on a straight road, while in Fig. 17c and 17d, the vehicle is travelling on a curving road. In all four images, the solid lane markers are unaffected by the averaging while the dashed lane markers give the appearance of a solid lane marker.

(a) $I(n)$.

(b) $I(n-3)$.

(c) $I(n-6)$.

(d) $I(n-9)$.

Figure 15: The four images used to create the Average Image.

Figure 16: Average Image $\bar{I}_K(n)$ that is created using $K = 3$ and $\Delta = 3$.

(a) Straight road.

(b) Straight road.

(c) Curving road.

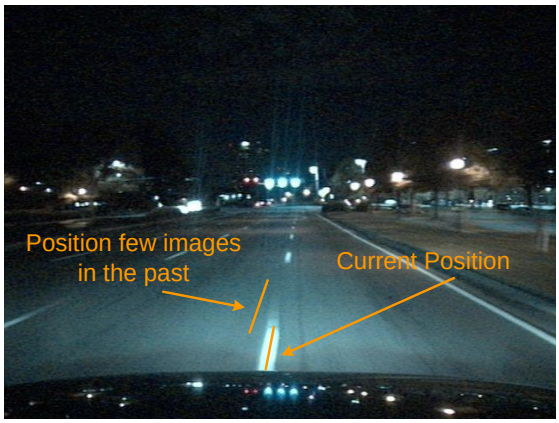
(d) Curving road.

Figure 17: Average Images computed on straight and curving roads.

However, there are occasions where the appearance of solid lane markers will be affected by averaging. For example, during a lane change, the solid lane markers may appear thicker in $\bar{I}_K(n)$ than they normally do as shown in Fig. 18. This happens because the lane markers in the previous images have moved laterally as shown in Fig. 19. As a result, when the images are averaged, the lane markers are not aligned; rather they are offset from each other giving a thicker appearance. However, the increased thickness is only a few inches in the Average Image. This can be explained with a simple example. Based on the FHA's specifications regarding the widths of a lane [19], the distance between the centers of the two lanes is estimated to be 12 ft. This is the distance that the vehicle needs to travel to complete a lane change as

Figure 18: Average Images computed on straight and curving roads.

(a) Image from the past.



(b) Current image.

Figure 19: Lane marker lateral movement during a lane change.

shown in Fig. 20. From our experiments, it was observed that a lane change during normal driving circumstances took an average of 8 seconds to complete. As a result, the lateral speed of the vehicle is 1.5 ft/s or 0.05 ft/33 ms. Since the oldest image used to compute $\bar{I}_K(n)$ is only 9 images in the past ($I(n-9)$), the lateral distance covered by the lane marker between $I(n)$ and $I(n-9)$ is $0.05 \times 9 = 0.45$ ft. We will return to this example towards the end of Sec. 2.4.2 where this number will make more sense. The dashed lane markers on the other hand appear as slanted line segments. However, this change in appearance of the dashed and the solid lane markers does not

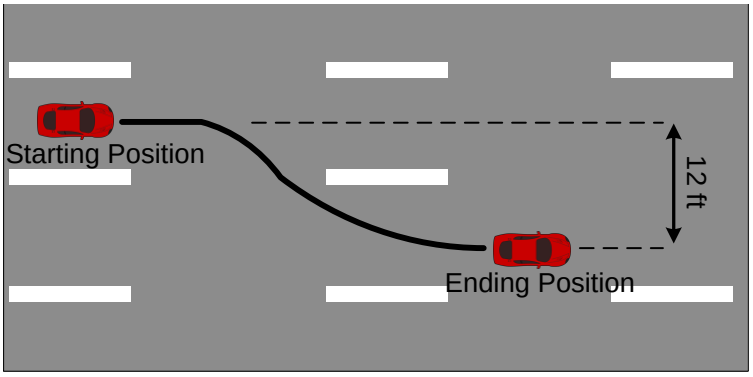


Figure 20: Illustration of a lane change.

seem to affect the performance of the ALD 1.0 as will be shown in the performance evaluation later on.

Finally, the Average Image $\bar{I}_K(n)$ is converted to a grayscale image $\hat{I}_n$ using a pixel-wise transformation [25] given by:

$$\hat{I}_n(r,c) = 0.299 \cdot R_n(r,c) + 0.587 \cdot G_n(r,c) + 0.114 \cdot B_n(r,c) \tag{2}$$

where $R_n(r,c)$, $G_n(r,c)$, and $B_n(r,c)$ are the red, green, and blue values of the pixel at location (r,c) in $\bar{I}_K(n)$ and $\hat{I}_n(r,c)$ is the intensity value of the pixel at (r,c) in the grayscale image $\hat{I}_n$ (notice the image index n is now in the subscript). Although $\hat{I}_n$ is made up of a single color channel while $\bar{I}_K(n)$ is made up of three color channels, both the white and the yellow lane markers are clearly visible in $\hat{I}_n$ as shown in Fig. 21. As a result, we can reduce the computational overhead of the system as the procedures executed by the next set of blocks in Fig. 12 will only need to be performed on a single color channel instead of multiple color channels.

2.4.2 Inverse Perspective Mapping

Next, Inverse Perspective Mapping (IPM) is applied to the grayscale image. IPM is a geometric transformation that removes the perspective effect from an image giving it the appearance of a bird's-eye view [20]. In comparison to the grayscale image in which the lane markers appear as lines that decrease in thickness and converge near

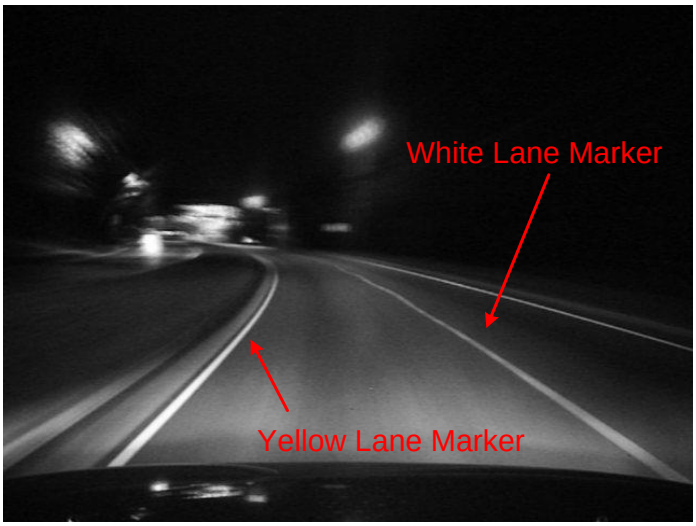
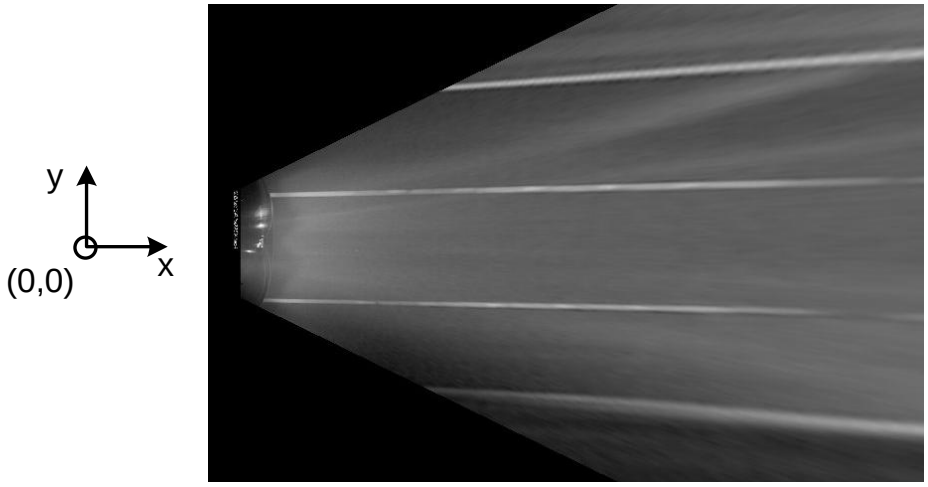


Figure 21: Yellow and white lane markers in the grayscale image $\hat{I}_n$.

the horizon, lane markers appear as near parallel lines that are of a constant thickness in an IPM transformed image [26]. A comparison between the grayscale image $\hat{I}_n$ and the IPM transformed grayscale image is shown in Fig. 22. By performing IPM on the grayscale image, lane detection now becomes a problem of locating a pair of near constant thickness parallel lines that are generally separated by a predetermined fixed distance in the image [26].

To create an IPM transformed image, we first have to establish a set of co-ordinate systems. The world co-ordinate system expressed as (x, y) in feet corresponds to the co-ordinate system of the IPM transformed image. The origin or optical center is located to the left and outside the image. The image co-ordinate system expressed as (r, c) in pixels corresponds to the co-ordinate system of the images prior to the geometric transformation is located at the top left of the image. The two co-ordinate systems are shown in Fig. 22. We will refer to the IPM transformed image as the "world image" and the pre-transformed image as the "camera image." In an IPM transformation, the mapping of each pixel (x, y) from the world image to the camera

(a) Camera image with origin on the top left.



(b) World image with origin on the left.

Figure 22: Comparison between camera and world images.

image (r, c) [27] is determined as

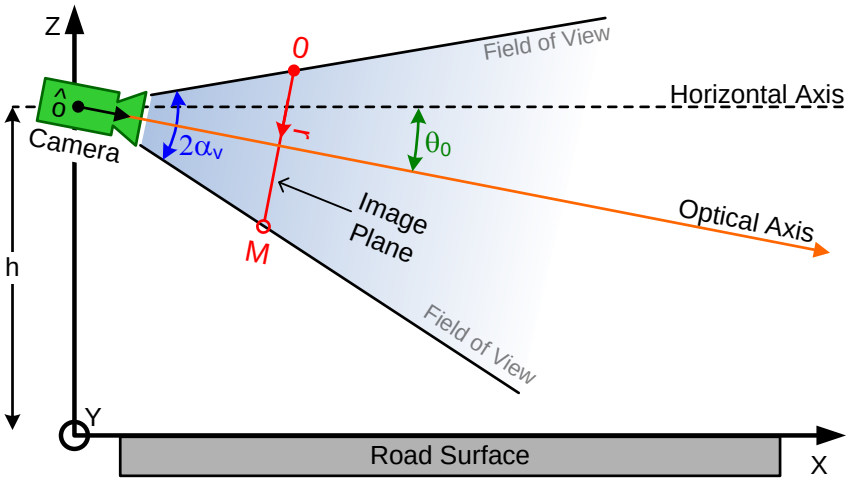
$$r(x) = \frac{M-1}{2}\left(1 + \frac{h - x' \cdot \tan\theta_o}{h \cdot \tan\theta_o + x'} \cdot \cot\alpha_v\right) + 1 \tag{3}$$

$$c(x,y) = \frac{N-1}{2}\left(1 - \frac{y'}{h \cdot \sin\theta_o + x' \cdot \cos\theta_o} \cdot \cot\alpha_u\right) + 1 \tag{4}$$

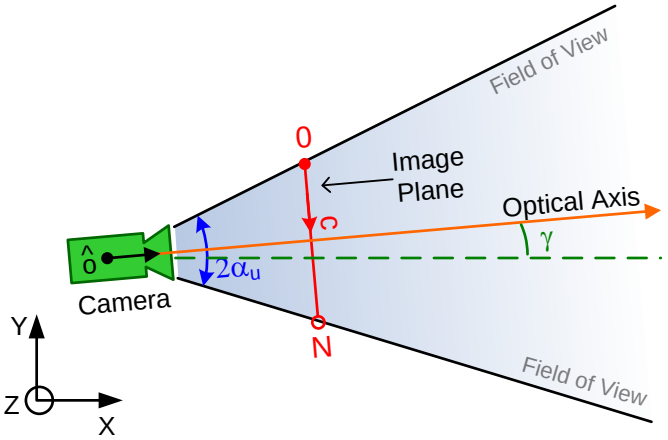
where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

In Eq. (3) and (4), h is the height of the camera from the ground, θ_o is the pitch below the horizon, α_u is the horizontal field of view, α_v is the vertical field of view, γ is the yaw, and $(M \times N)$ is the size of the camera image. In Fig. 23a, a side-view or a view along the Y-axis shows the camera configuration along and its related calibration parameters with $\widehat{o}$ a unit vector representing the camera's optical axis. Similarly, in



(a) Along the Y-axis (side-view).



(b) Along the Z-axis (top-view).

Figure 23: Camera configuration.

Fig. 23b, the camera configuration is shown along the Z-axis or a top-view. The values of α_u and α_v used in Eq. (3) and (4) are provided by the lens manufacturer. Similarly, the values of M and N are provided by the camera manufacturer. θ_o was determined by estimating the location of the horizon (r_H) in the camera image as

shown in Fig. 24 and then solving

$$\theta_o = \tan^{-1}\left(\left[1 - 2\left(\frac{r_H - 1}{M - 1}\right)\right]\tan\alpha_v\right) \tag{6}$$

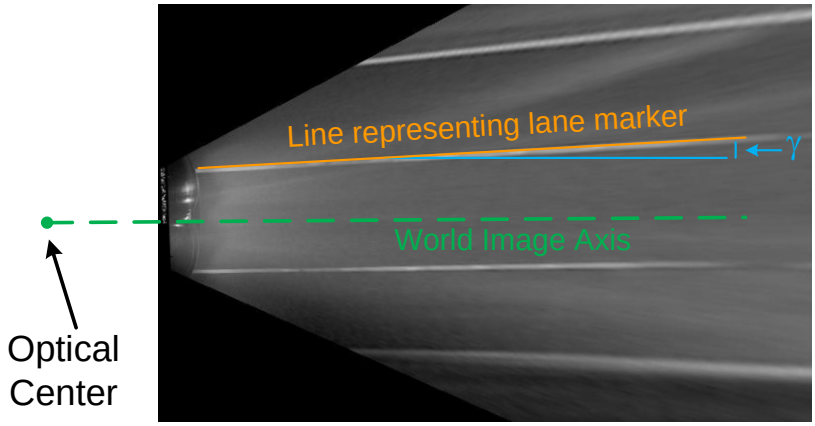
as shown in [27].

Figure 24: Location of the horizon (r_H) in the camera image.

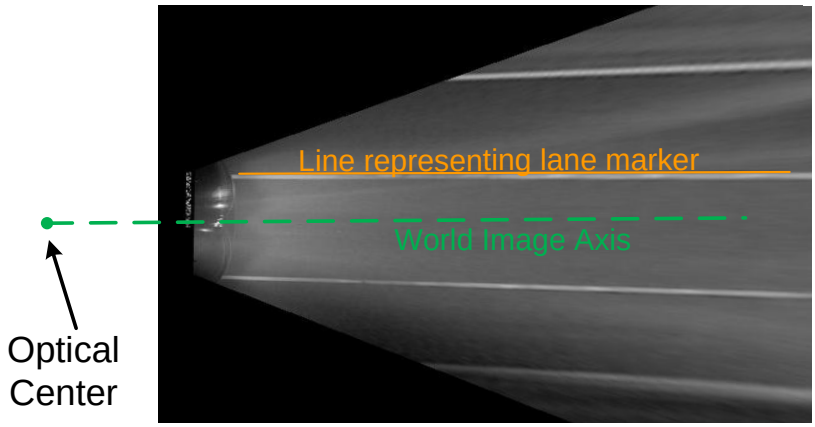
Assuming that the vehicle is traveling on a straight road with the straight lane markers as shown in Fig. 25, γ was determined by rotating the world image about the camera's optical center until the lane markers in the image were horizontal.

Since the world image is perspective free, the distance in the real world that corresponds to the distance between two pixels is constant over the entire image. For example, the world image in the IPM block is created with a resolution of 8 pixels/foot. This means that the distance between 8 pixels in either the horizontal or the vertical directions of the world image equates to one foot in the real world. This inter-pixel distance relation does not exist in the camera image because it is not free of perspective. The 8 pixels/foot resolution is selected so that the lane markers have a thickness of a few pixels in the world image rather than appearing as thin line segments as seen in Fig. 22b.

For the IPM transform to be effective, the road surface is assumed to be flat in the vicinity of the vehicle and θ_o remains fixed [20]. At times when θ_o changes e.g.



(a) World image containing some yaw γ.

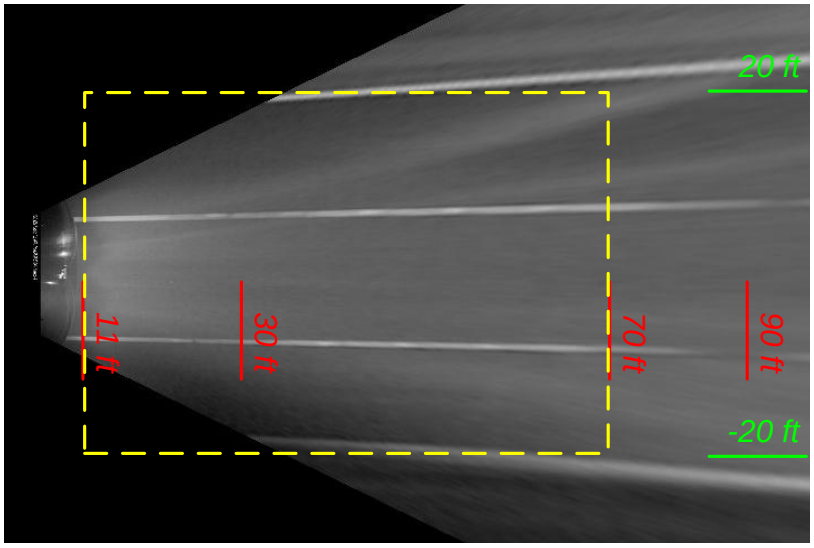


(b) Yaw corrected world image.

Figure 25: Gamma correction procedure in the world image.

when the vehicle encounters a bump, or the vehicle enters an inclined surface such as a slope, the parallel lane markers in the world image will appear skewed inwards or outwards. Fortunately, the camera mount used in our vehicle is sturdy and keeps θ_o fixed at almost all times. Additionally, the changes in the inclination were only temporary and had only a little effect on the performance of the lane detector as will be shown in Sec. 2.4.9. An interesting observation that can be made about the world image is the blur towards the right of the image. This blur is introduced by interpolation since large portions of the road surface are compacted into a few rows in the camera image as we move towards the horizon. This is illustrated in Fig. 26a. As a result, we only utilize the pixels up to a distance of 70 ft in the world IPM for

(a) The locations of various distances ahead of the vehicle in the camera image.



(b) The locations of various distances ahead of the vehicle in the world image. The area utilized for further processing is shown by the dotted yellow rectangle.

Figure 26: Distances in the camera and world image.

further processing. Finally, a world image is created with dimensions of 321×431 pixels where $x \in [11, 70]$ ft and $y \in [20, -20]$ ft and is represented by the dotted yellow rectangle in Fig. 26b. The horizontal and the vertical spacing between adjacent pixels in the world image is 0.125 ft or 1/8 ft.

Before we conclude this section, let us refer back to the example in Sec. 2.4.1.

We had mentioned that the lane markers move 0.45 ft between 9 images during a lane change. Based on the resolution of the world image discussed previously, 0.45 ft $\times$ 8 pixels/foot $=$ 3.6 pixels in the world image. Therefore, the solid lane markers will only appear a few pixels thicker in the perspective free image as shown in Fig. 27. This phenomenon did not seem to affect the performance of the lane detector as will be shown in the performance evaluation section.

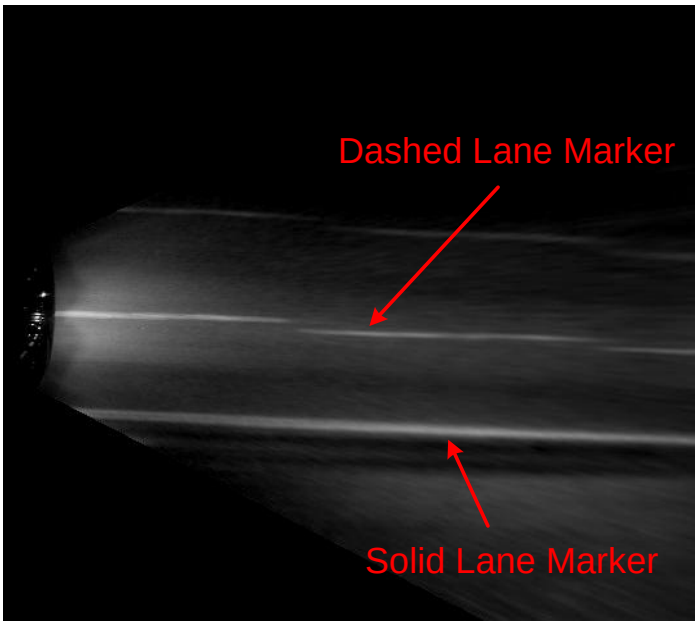


Figure 27: Lane markers as they appear in a world image during a lane change.

2.4.3 Object of Interest Detection

The focus of this block is to transform the world image into a binary image in which the lane markers are detected. This is done by applying an adaptively computed threshold to each pixel in the world image [24]. The algorithm for computing the threshold value for each pixel is shown in Algorithm 1. We start by creating a blank binary image B_n that has the same dimensions as the world image. For simplicity, we will refer to the world image as W_n. Furthermore, a pixel at location (x, y) in W_n will be denoted as $W_n(x, y)$; similarly, $B_n(x, y)$ is a pixel at location (x, y) in the binary image. Then, for each pixel in $W_n(x, y)$, we create a $k \times 1$ window centered at

(x, y) and determine the difference between the minimum and maximum intensities

Algorithm 1 Adaptive Threshold [28]

for Each Pixel (x,y) **do**
  min = Minimum value in the neighborhood around $W_n(x, y)$
  max = Maximum value in the neighborhood around $W_n(x, y)$
  if $(max - min) > \alpha$ **then**
    T = Mean of values in the neighborhood around $W_n(x, y)$;
    if $W_n(x, y) > T$ **then**
      $B_n(x, y) = 1$
    else
      $B_n(x, y) = 0$
    end if
  else
    $B_n(x, y) = 0$;
  end if
end for

within the window. If the difference does not exceed α, then the pixel at $B_n(x, y)$ is set to 0 implying that there is not much intensity variation within the window. If the difference exceeds α, then there is sufficient intensity variation and part of a lane marker probably exists within the window. In Fig. 28, we have shown the difference

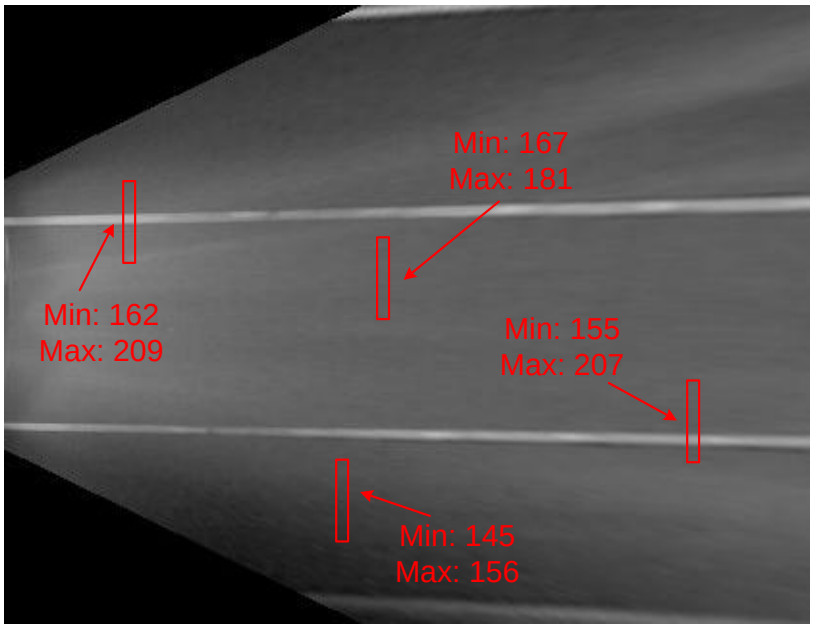
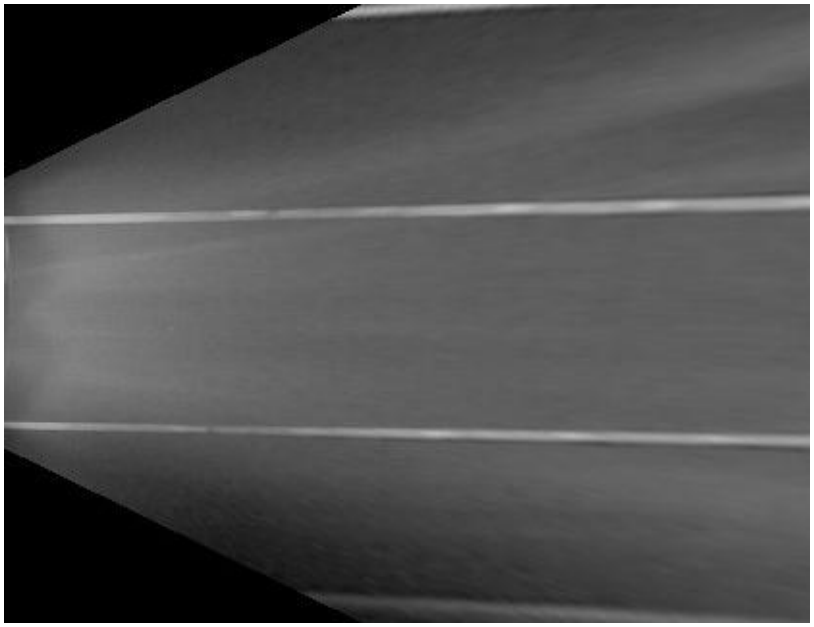


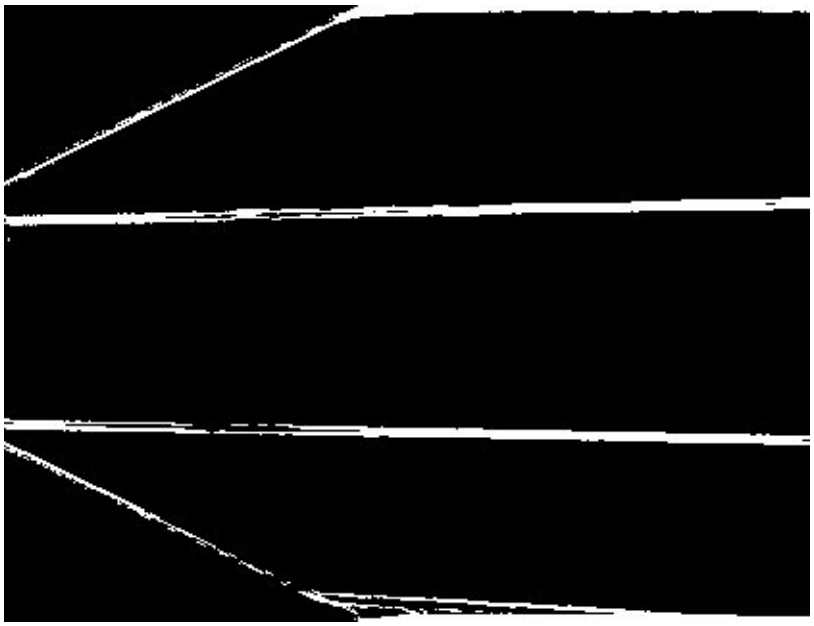
Figure 28: Statistics within several windows in W_n.

in min/max values for the different locations of the $k \times 1$ window. Next, a threshold T is set to the arithmetic mean of the intensities within the window and the intensity value at $W_n(x, y)$ is compared to T. Here, we observed that if the intensity value at $W_n(x, y)$ was greater than T, then $W_n(x, y)$ was probably a part of the lane marker; otherwise, it was a part of the road surface. Therefore, if $W_n(x, y)$ is less than T, then $B_n(x, y)$ is set 0; otherwise, $B_n(x, y)$ is set to 1. In Fig. 29, we have shown the world image W_n and its corresponding binary image B_n. By setting the shape of the window to a column vector, the adaptive threshold algorithm is made sensitive to the intensity changes along the vertical direction. As a result, lane markers are detected since they are predominantly horizontal. However, there were a few occasions where the sides of other cars or trucks were also detected in the binary image. One example of such a case is shown in Fig. 30. Furthermore, the values of k and α could not be determined analytically; therefore, we tested k and α with several empirically determined values and observed that $k = 5$ and $\alpha = 7$ produced the best result. We will show the performance for each combination of k and α in Appendix A.

When a global threshold is computed using the pixels values of the entire image, the computation of an ideal global threshold becomes difficult when remote bright objects are present in a relatively dark image. On the other hand, the method described above generates a unique threshold value for each pixel which is determined by the characteristics of a small set of pixels within a window [28]. As a result, the presence of remote bright objects such as headlights and taillights of cars do not affect the threshold calculation of pixels in different parts of the image. Consequently, this method proves advantageous over the global threshold when creating a binary world image.



(a) Grayscale world image W_n.



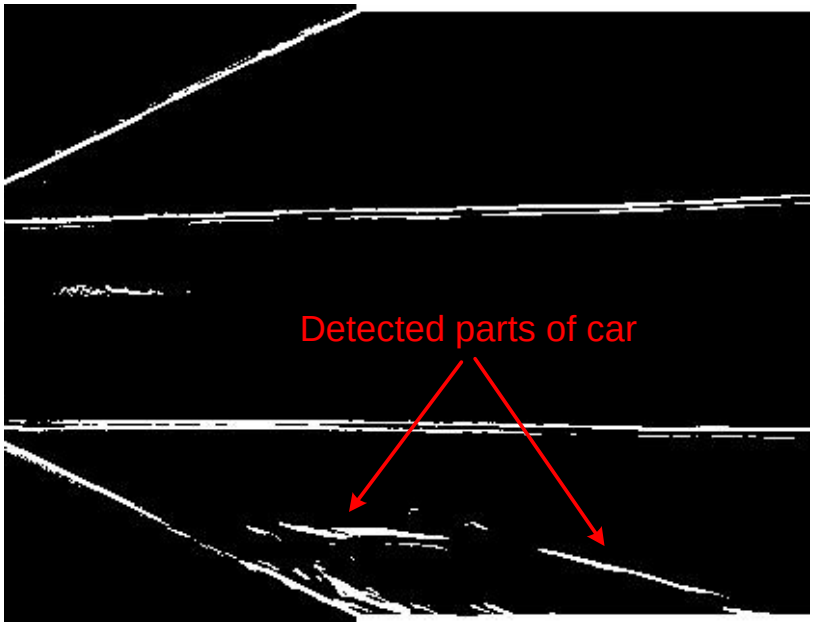
(b) Binary image B_n.

Figure 29: Converting the world image W_n to a binary image B_n.

2.4.4 Preliminary Feature Extraction

The goal of this block is to determine rough estimates of the locations of lane markers in the world image W_n with the assistance of the binary image B_n [24]. As shown in Fig. 31, this block is divided into three stages:

(a) Camera image with nearby car.



(b) Parts of car detected in B_n.

Figure 30: Artifacts detected in the binary image.

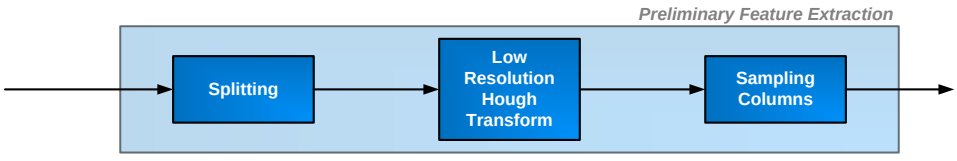


Figure 31: Stages in the preliminary feature extraction block.

1. Splitting

2. Low Resolution Hough Transform

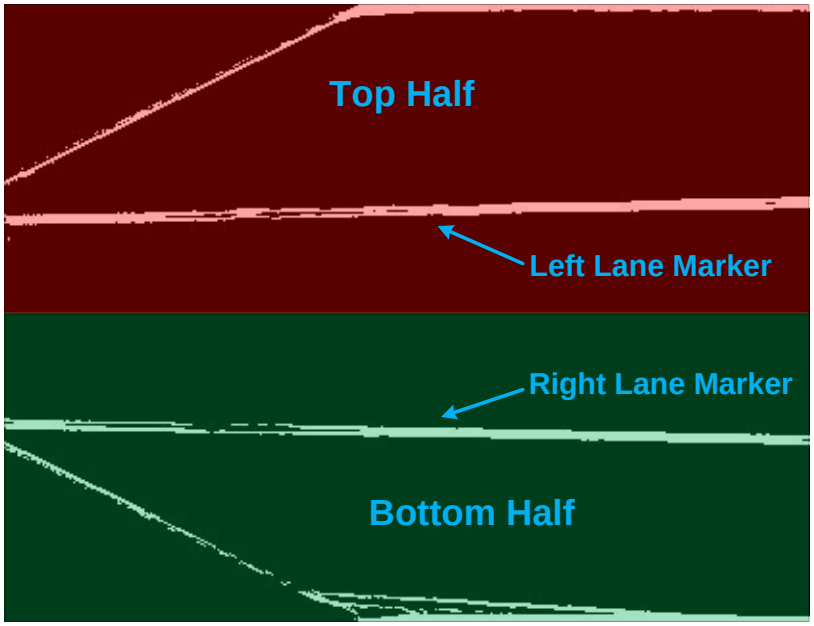
3. Sampling Columns

In splitting, the binary image B_n is split along the middle into two halves: a top half, and a bottom half. The reason for splitting the image is because we observed that the lane markers for the left lane boundary often appear only in the top half image. Similarly, the lane markers for the right lane boundary appear in the bottom half image. We have shown two cases in Fig. 32 that illustrates this condition. Furthermore, the detections of the left and the right lane markers are not dependent on each other; hence, they can be performed separately. For illustration, we will only use the bottom half image from now on and refer to it as H_n.

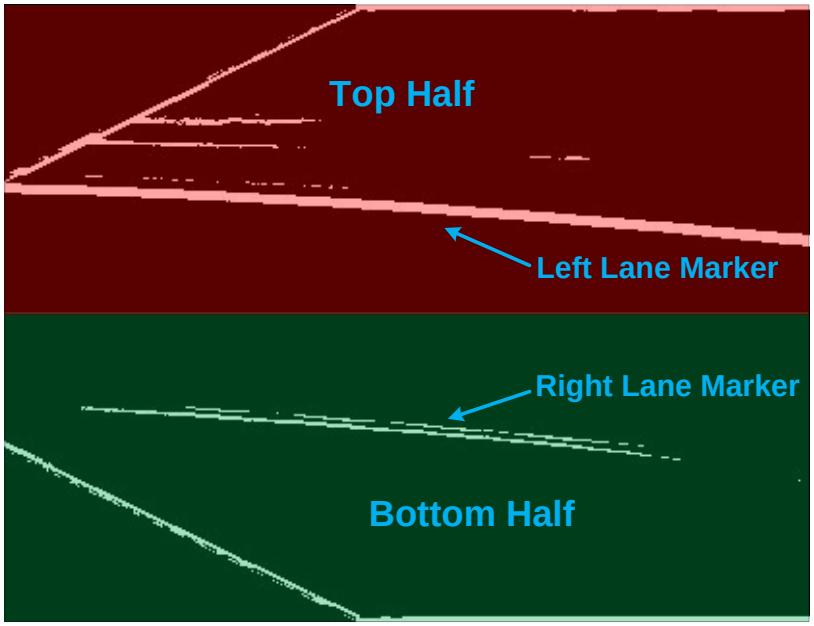
In H_n, it can be observed that the lane markers appear as white line segments. Therefore, detecting the locations of these line segments is useful. A popular method for line detection is the classical Hough Transform (HT) [22]. Let us take a look at a simple example of how the Hough Transform works. Consider a binary image that contains a few non-zero pixels that lie along a straight line as shown in Fig. 33. A straight line can be described in the normal form as

$$\rho = x_i \cdot \cos\theta + y_i \cdot \sin\theta \tag{7}$$

where (x_i, y_i) are the co-ordinates of the non-zero pixels, ρ is the perpendicular distance between the line and the origin (top left), and θ is the angle formed by the perpendicular and the origin as shown in Fig. 33. The goal of the Hough Transform is to determine the values of ρ and θ that best describe the line. The classical HT begins with an empty parameter space called the "Hough space." Specific entries in the Hough space correspond to specific values of ρ and θ. Then, for each non-zero pixel i located at (x_i, y_i) in the binary image, θ sweeps the interval $[-90, 90)$. At



(a) On a straight road.



(b) On a curving road.

Figure 32: Lane markers that appear in the top (red overlay) and bottom (blue overlay) half of the image.

each value of θ, the corresponding value of ρ is determined by Eq. (7) and the $(\rho, \theta)^{\text{th}}$ element in the Hough space is incremented by one. After iterating over all non-zero pixels in the image, the Hough space will appear as shown in Fig. 34. In Fig. 34,

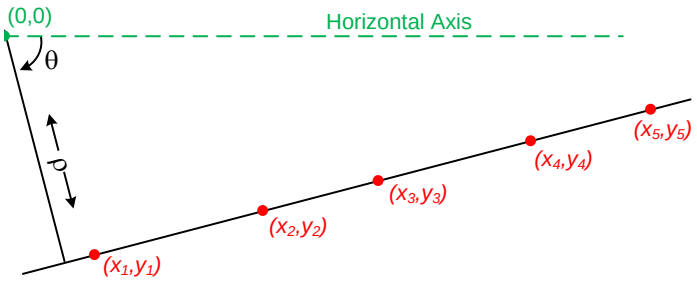


Figure 33: Points that lie on straight line.

the white spot is a peak and corresponds to the (ρ, θ) pair that best describes the line in the image. However, most images in this application will not contain a single line. For example, in Fig. 29a and 29b, multiple lines exist in the images. Therefore, the corresponding Hough spaces for these images will contain multiple peaks, one for

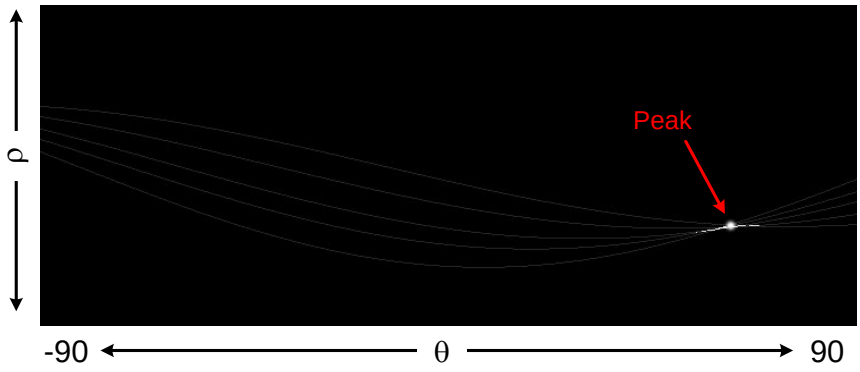


Figure 34: Hough space after computing the Hough transform.

each line in the image. Finally, the value at each $(\rho, \theta)^{\text{th}}$ element in Hough space corresponds to the number points that lie on the line described by (ρ, θ). Therefore, the index of a peak contains a large value because numerous points lie along the line described by that particular (ρ, θ) pair.

Based on the discussion above, it is clear that the Hough Transform is very effective in finding lines of any orientation in the image. However, in this work, the lines corresponding to the lane markers are mostly horizontal. Therefore, the classical Hough Transform spends time searching for lines of orientations that may not exist in the image or otherwise correspond to possible false positives such as pedestrian

crosswalks or bumpers of vehicles. Hence, the classical approach needs to be modified. This is where the Low Resolution Hough Transform (LRHT) comes in. The LRHT is an optimized version of the classical Hough Transform that is used to find near horizontal lines by restricting the sweep interval for θ [29]. For example, in the classical approach, θ sweeps the interval $[-90, 90)$. In the LRHT, θ only sweeps the interval $\{[-90, -75], [75, 90)\}$. In addition, the increments between each θ value in the classical approach are often $0.25°$ or $0.5°$; whereas in the LRHT, the increments are $1°$ or $2°$. Therefore, if we consider an image containing 10 non-zero pixels that lie on a near horizontal line, the classical approach will compute 3600 θ values (for $0.5°$ increments) whereas the LRHT will only compute 310 θ values (for $1°$ degree increments). As a result, the LRHT is much faster that the classical approach. However, due to the smaller θ increments, the classical approach is able to detect the (ρ, θ) parameters of the lines more accurately than the LRHT. But, the LRHT is able to produce rough estimates for these parameters that are sufficient for this block. As a result, the LRHT is used to find line segments that represent lane markers in H_n. The Hough space that is created using the LRHT is shown in Fig. 35. Since

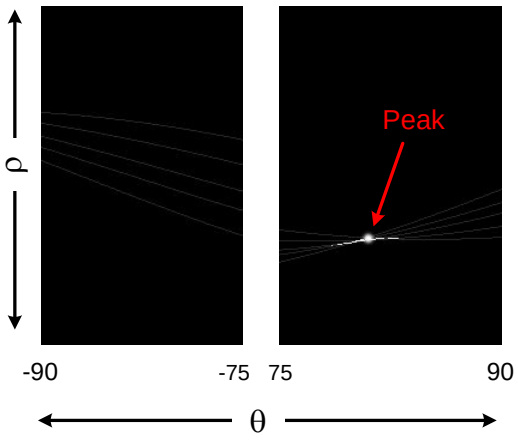
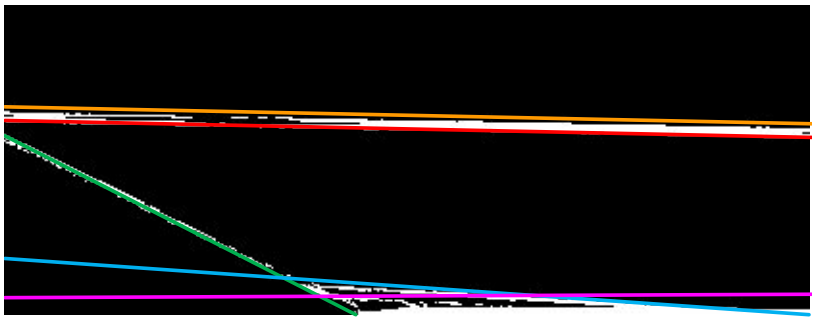


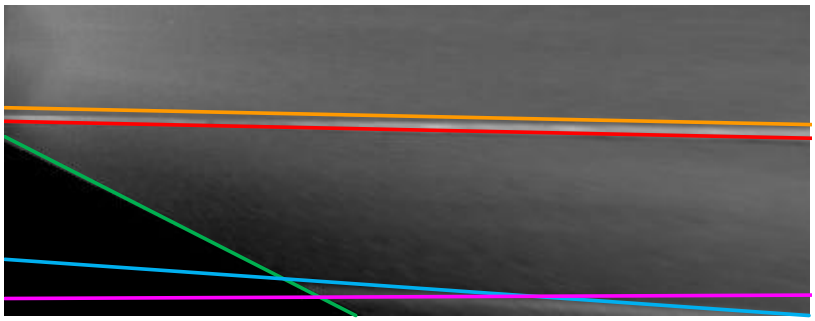
Figure 35: Hough space.

multiple lines may exist in the images, we select at most ten lines that correspond to the ten highest largest (ρ, θ) elements in the Hough space. We do this because

it is very likely that the largest (ρ, θ) element may not represent the lane markers; although, it is very likely that it is among the largest ten. Therefore, we select ten lines. Furthermore, each of the ten elements needs to exceed a value of 50 implying that at least 50 pixels lie on each line. This is a conservative requirement considering that H_n contains $321 \times 431/2 \sim 70,000$ pixels. In Fig. 36a, we show five lines that represent the only five peaks in Hough space that exceed the value of 50.



(a) Five lines in H_n that represent the five largest peaks in the Hough space.

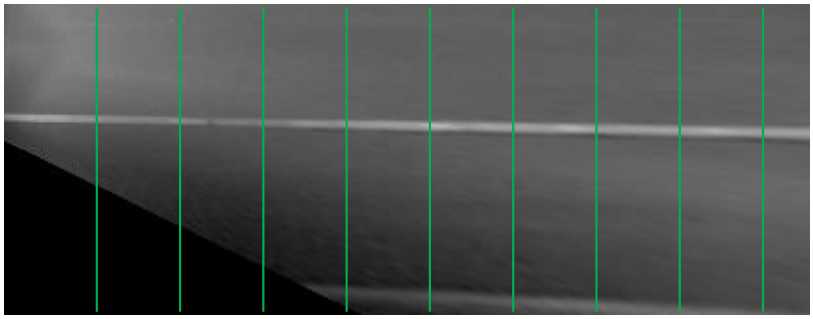


(b) Lines from H_n that are now overlaid in H'_n.

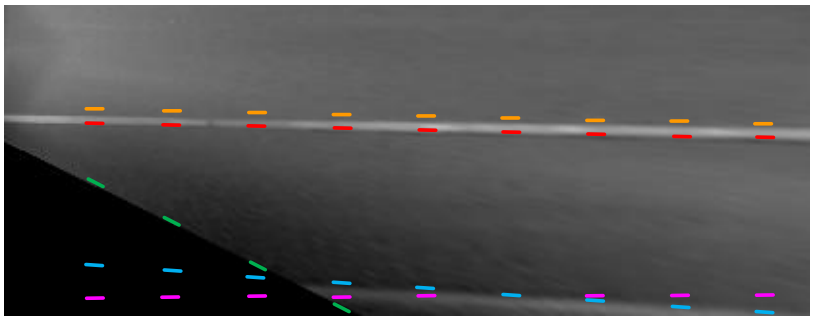
Figure 36: Transfer of lines from H_n to H'_n.

Next, we create an H'_n, the bottom half image of the world image W_n. Then, we map the five lines from H_n to H'_n. Since there is no geometric transformation between H_n and H'_n, the lines can simply be transferred and overlaid. The grayscale image H'_n overlaid by the lines is shown in Fig. 36b. Note that, there are times where only the borders of the lane markers may be detected and not the entire lane marker as shown in Fig. 32b (only the border of the right lane marker is detected); therefore, further processing is performed in H'_n. Next, we set up Sampling Columns.

Sampling Columns are the columns in H'_n where each of these lines will be sampled. The purpose of the Sampling Columns is to convert the lines into a discrete set of points, and these points after some modifications will be used to estimate the lane boundary curve in the later blocks. The Sampling Columns are spaced 24 pixels apart which is the equivalent of 3 ft in the real world and are shown in green in Fig. 37a. The discrete sets of points for each line are shown in Fig. 37b. Moreover, these points serve as the preliminary or rough locations of the lane markers in H'_n.



(a) Sampling Columns in H'_n.



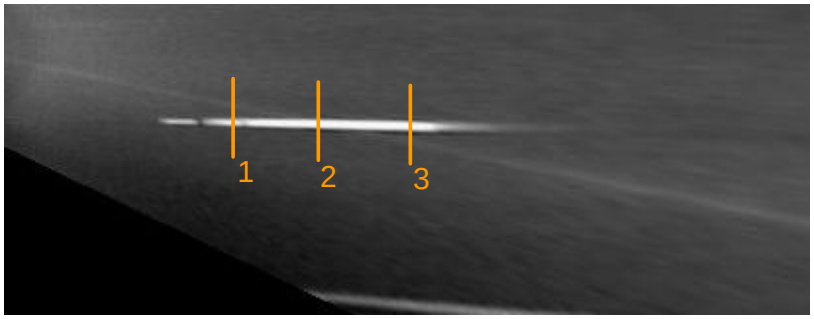
(b) Discrete points that represent the lines sampled in the Sampling Columns.

Figure 37: Sampling the lines into a discrete set of points.

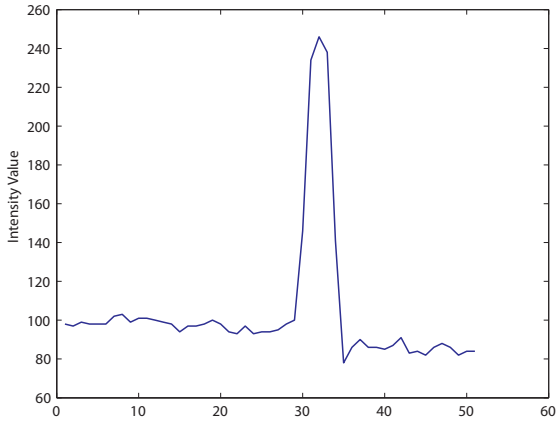
2.4.5 Detailed Feature Extraction

The goal of this block is to determine the exact location of the lane markers in each Sampling Column shown in Fig. 37b. This location is determined through a series of template matching procedures that are subject to threshold requirements. For a solid lane marker, a location will most likely exist in each column. However, due to

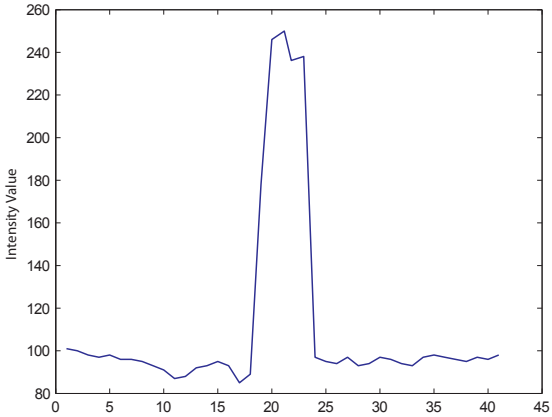
the gap between the dashed lane markers, it is possible that some columns may not contain a lane marker.



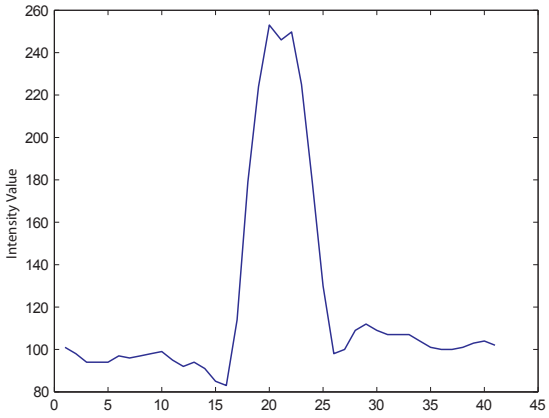
(a) Three ROIs in the world image are represented by the vertical orange bars.



(b) Intensity profile of the lane marker inside ROI 1.



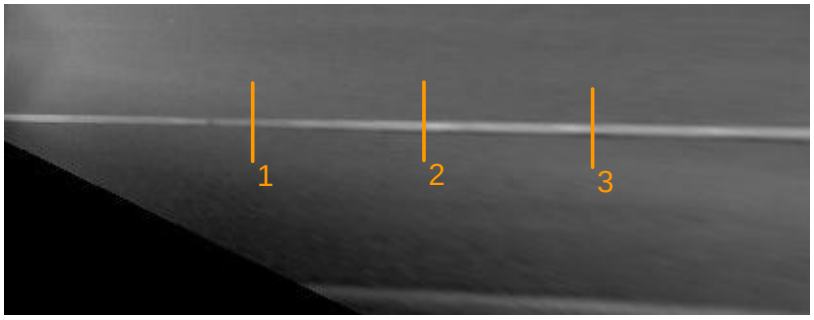
(c) Intensity profile of the lane marker inside ROI 2.



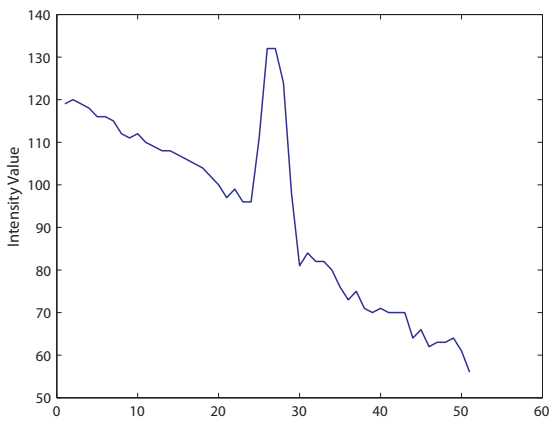
(d) Intensity profile of the lane marker inside ROI 3.

Figure 38: Intensity profile of a lane marker within different ROIs.

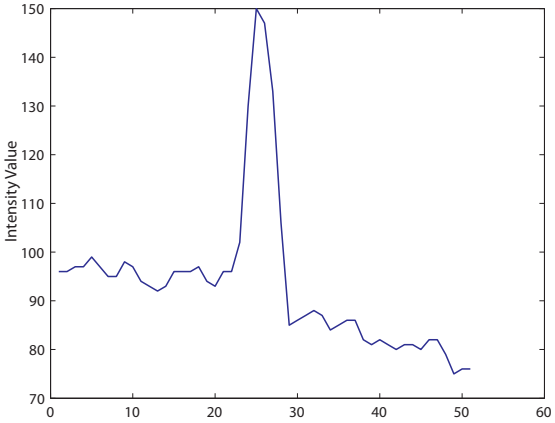
Now, with a collection of preliminary lane marker locations produced in the previous block, it is essential to check the validity of each point or location. We do this by checking the one dimensional (1-D) intensity profile in the neighborhood of each point. To elaborate, if we move from the top to the bottom of H'_n along one of these Sampling Columns, then, in the area that contains the lane markers, the intensity profile will ideally appear in the shape of a rectangular pulse. However, due to the noise in the image, intensity variations, contrast differences between the road surface and the lane marker etc. this rectangular pulse often appears as a peak. In Fig. 38, we have shown the intensity profiles of a lane marker in three selected regions of the world image. These selected regions or Regions of Interest (ROI) are represented by the three orange bars in Fig. 38a and are numbered 1, 2, and 3. Furthermore, H'_n contains temporal blurring which is the result of averaging several images from the past. The past images often contain some lateral lane marker movement; as a result, the peak is smoothened and has a shape similar to a bell or Gaussian curve [24]. We have shown the bell shaped intensity profile for a few lines in a few images in Fig. 39 and 40. Since a normal lane marker is 6 inches wide in the real-world or 4 pixels wide in H'_n, the template is created as a 51 pixels wide Gaussian shaped curve and two standard deviations equal 2 pixels. This specification for the standard deviation allowed us to create a template that most closely resembles the intensity profile of a lane marker as shown in Fig. 39 and 40. However, as mentioned in the introduction, the normal, the wide, and the double lane markers are most commonly encountered on the road and in H'_n. Therefore, we need to create templates for the wide and double lane markers as well. The template for the wide lane marker is created in the same manner as the template for the normal lane marker except that two of its standard deviations equal 5 pixels to adjust for the 10 inch width of the wide lane marker in the real-world. Similarly, the template for a double lane marker is created as a combination of two normal lane marker templates and the means are separated



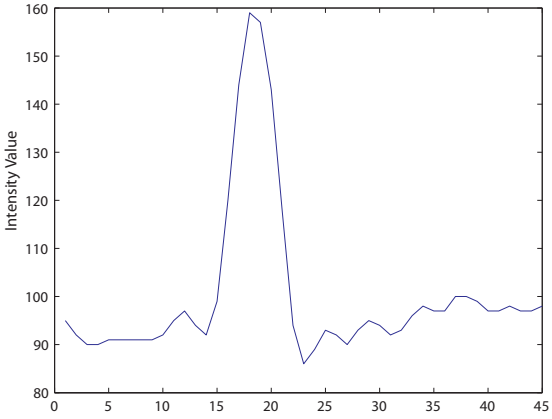
(a) Three ROIs in H'_n of a straight road are represented by the vertical orange bars.



(b) Intensity profile of the lane marker inside ROI 1.



(c) Intensity profile of the lane marker inside ROI 2.

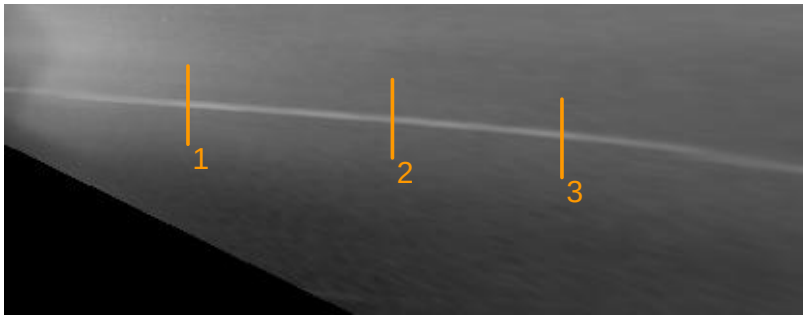


(d) Intensity profile of the lane marker inside ROI 3.

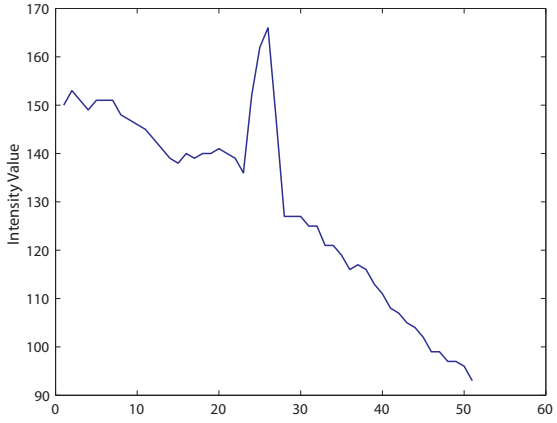
Figure 39: Intensity profile of a lane marker within different ROIs.

by a 4 pixels in H'_n or 6 inches in the real world. As a result, we now have a set of the templates as shown in Fig. 41 that can be used for template matching.

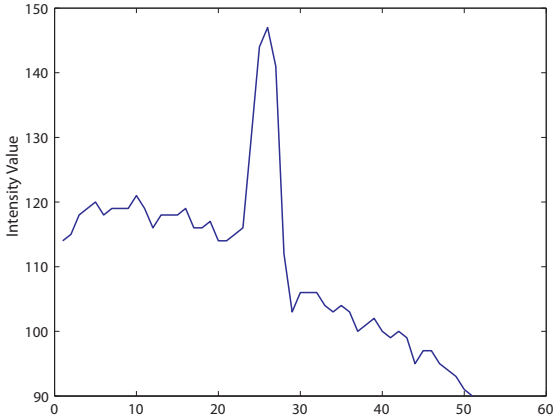
Template matching by Normalized Cross Correlation (NCC) [30] is now performed



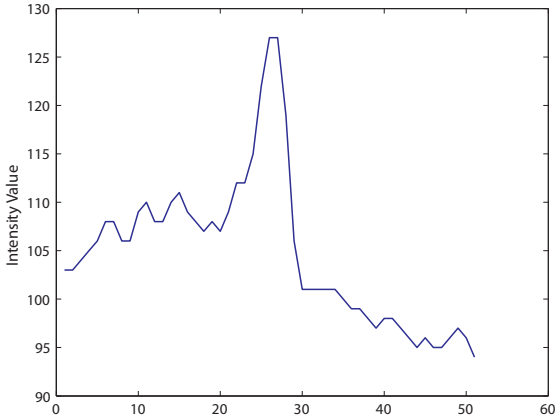
(a) Three ROIs in H'_n of a curving road are represented by the vertical orange bars.



(b) Intensity profile of the lane marker inside ROI 1.



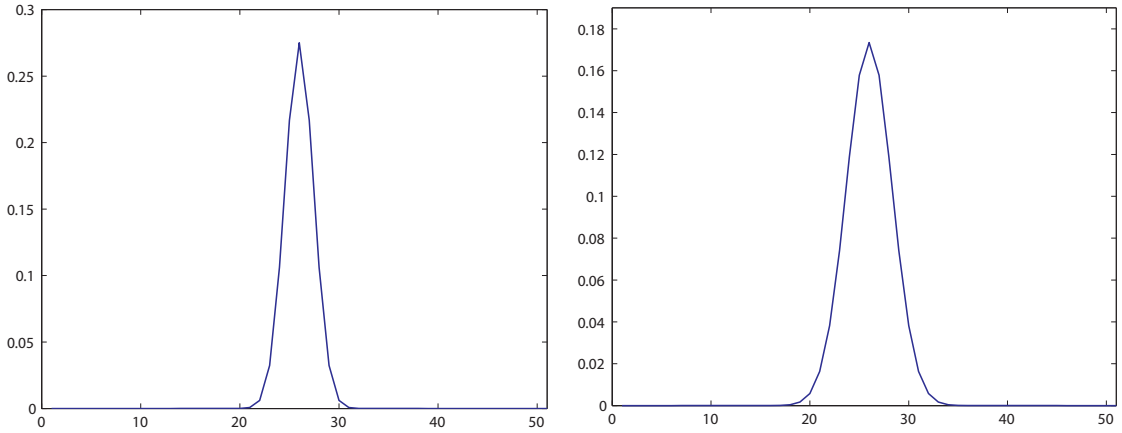
(c) Intensity profile of the lane marker inside ROI 2.



(d) Intensity profile of the lane marker inside ROI 3.

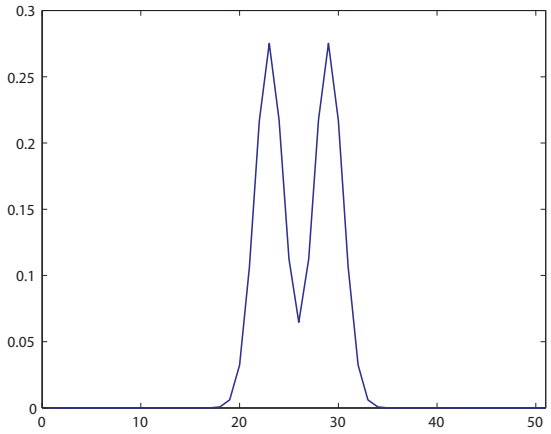
Figure 40: Intensity profile of a lane marker within different ROIs.

in 1-D in each Sampling Column using each of the three templates described earlier. In this application, NCC is used to determine the similarity between the template and the areas along the Sampling Columns. However, NCC is not performed along



(a) Template of a normal lane marker.

(b) Template of a wide lane marker.



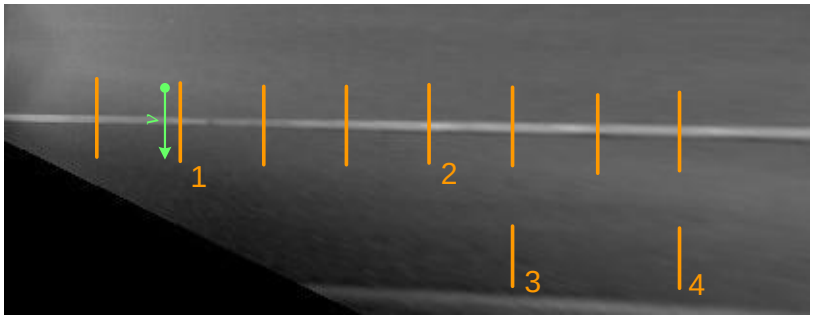
(c) Template of a double lane maker.

Figure 41: Set of templates to be used in template matching.

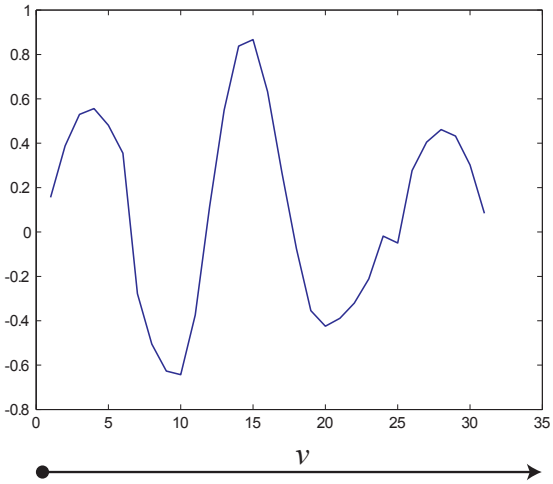
the entire Sampling Column (top to bottom), rather only in an ROI around each preliminary lane marker location in Fig. 42a [24]. As before, the ROIs are represented by vertical orange bars and extend 16 pixels (2 ft) above and below each location as shown in Fig. 42a. The NCC coefficient is calculated as

$$\gamma(v) = \frac{\sum_y [f(y) - \overline{f}_v][t(y-v) - \overline{t}]}{\sqrt{\sum_y [f(y) - \overline{f_v}]^2 \sum_y [t(y-v) - \overline{t}]^2}} \tag{8}$$

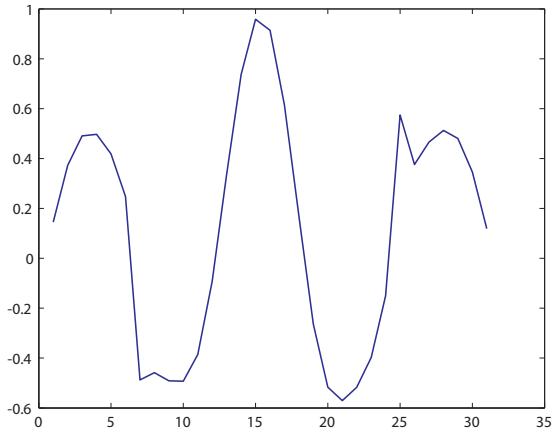
where $\gamma(v) \in [-1, 1]$ and v is a location within the ROI as shown in Fig. 42a. NCC coefficient computed using the normal lane marker template within the different ROIs are shown in Fig. 42b−42e. The reason for setting up the neighborhood is because the preliminary locations of the lane markers provided by the previous block are



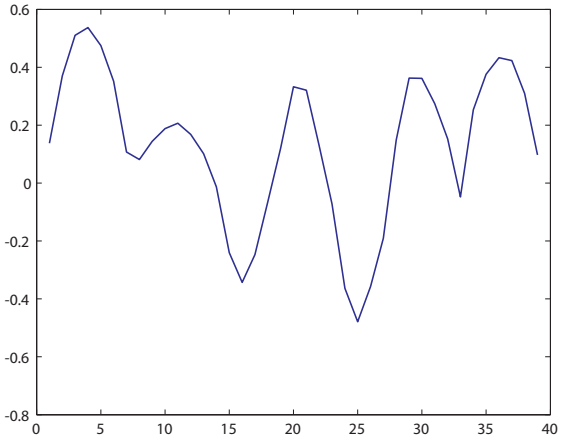
(a) H'_n with multiple ROIs.



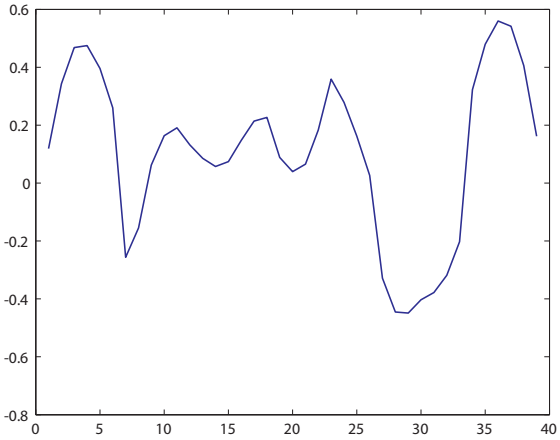
(b) Correlation coefficients inside ROI 1. The location of v from (a) is shown at the bottom of plot.



(c) Correlation coefficients inside ROI 2.



(d) Correlation coefficients inside inside ROI 3.



(e) Correlation coefficients inside ROI 4.

Figure 42: Correlation coefficients within the different ROIs.

preliminary estimates as stated before; therefore, they may not be aligned with the center of the lane marker in that particular Sampling Column. For example, on a

curving road, the preliminary estimates may not be aligned with the centers of the lane markers in each Sampling Column. As a result, by template matching within the neighborhood of each preliminary location, the center of the lane marker is estimated. Additionally, the neighborhood restricts the template matching only to the vicinity of the preliminary locations to avoid possible false signaling from other areas along the Sampling Columns. Within the neighborhood, the location of the maximum value of the NCC coefficient often corresponds to the centers of the lane markers as shown in Fig. 42b and 42c. However, there are times e.g. in ROI 3 and ROI 4 where NCC does not produce very large coefficients. In these situations, simply selecting the location that corresponds to the maximum value in the coefficient map may not necessarily correspond to a lane marker as shown in Fig. 42d and 42e. Therefore, the coefficient map is also subject to a threshold. However, since template matching is performed with three templates, all three coefficients maps that are produced as a result are subject to the threshold. Basically, the location that corresponds to the maximum value in the three coefficients maps and also exceeds a threshold τ is considered the center of the lane marker. The threshold was determined by collecting the NCC coefficients from 80 positive and 80 negative training samples and then setting τ to the cut-off point on the Receiver Operator Characteristics (ROC) curve that produced the highest accuracy. The ROC curve is a plot of the sensitivity of a binary classifier system for different threshold values [31]. To create the 80 positive training samples, we took 80 intensity profiles containing the lane markers, some of which are shown in Fig. 39b−39d and 40b−40d. Each profile was template matched with its respective template i.e. profile of a normal lane marker was template matched with a normal lane marker template, similarly, template matching was performed between the wide and the double lane markers as well. Then, the largest value in the correlation map was collected for each of 80 template matches. The 80 negative training samples were similarly created except that the intensity profiles were taken from portions of the

road that did not contain lane markers. In Fig. 43, we have presented the ROC curve for the 160 training instances. We have also provided a confusion matrix in Table 1 containing the predicted outcomes for the 160 training instances that were produced by setting the threshold $\tau = 0.739$. This threshold yielded the highest accuracy. The accuracy (ACC) is computed as

$$ACC = \frac{TP + TN}{P + N} \tag{9}$$

where TP is the number of true positives, TN is the number of true negatives, P is the number of positive training instances, and N is the number of negative training instances. Based on the data provided in Table 1, TP= 68, TN= 70, and P=N= 80.

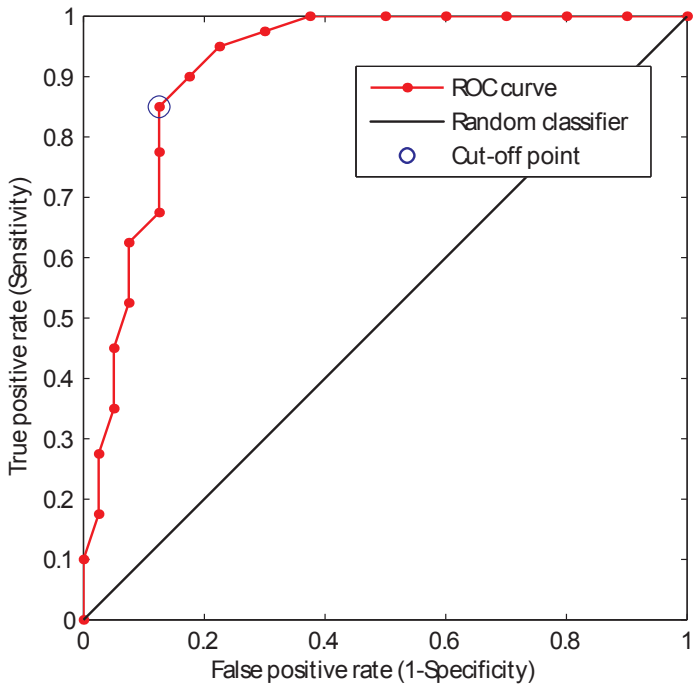


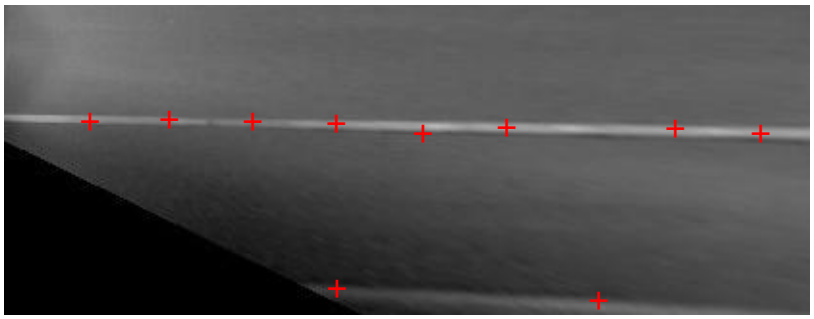
Figure 43: ROC curve for the 160 training instances.

The template matching process described above is repeated for each preliminary lane marker location in each line provided by the previous block. If multiple lines exist in H'_n and they are spread out, it is possible that multiple locations in each Sampling Column may be considered as lane marker centers. An example of such

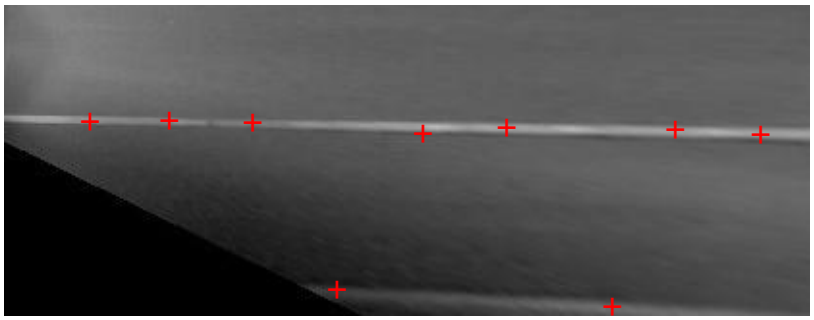
Table 1: Confusion matrix for the 160 training instances. $\tau = 0.739$, Accuracy (ACC)= 0.863, Area under the ROC curve (AUC)= 0.917, True Positive Rate (TPR)= 0.85, Specificity (SPC)= 0.875.

		Predicted Outcome		
		Positive	Negative	Total
Input	Positive	68	12	80
	Negative	10	70	80
	Total	78	82	160

a situation is shown in Fig. 44a. Then, we only select the point in each Sampling



(a) Multiple locations considered as lane markers in a Sampling Column.



(b) Location corresponding to the largest correlation coefficient in each Sampling Column is considered.

Figure 44: Locations of lane markers in the Sampling Columns.

Column that produced the largest correlation coefficient. Upon completion, we have the location of the lane marker centers in several Sampling Columns as shown in Fig. 44b.

2.4.6 Data Modeling

It is now possible to estimate the boundary of the lane using the points or locations of the lane marker centers in the Sampling Columns. This is done by modeling a curve that serves as a best fit to these points. However, it is not advisable to try to model a curve directly to all the points because there could be outliers. Fig. 45 shows an example of the inliers (red) and the outliers (blue) that are present in H'_n. If we try to fit either a straight line or quadratic curve to all these points, it is very likely that the curve will not have the best representation in the presence of the

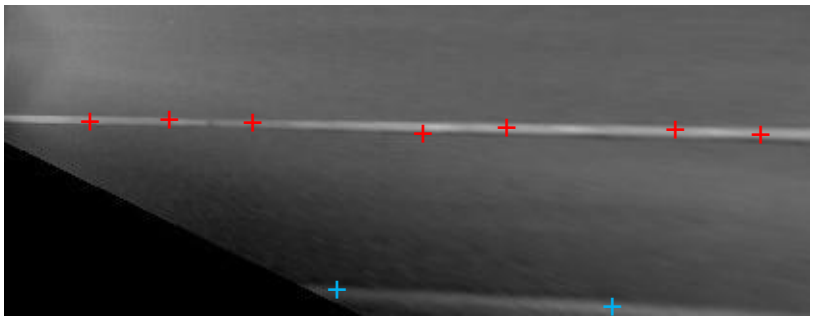
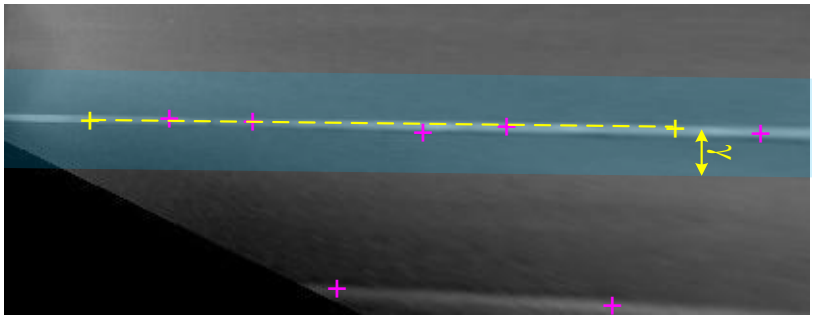


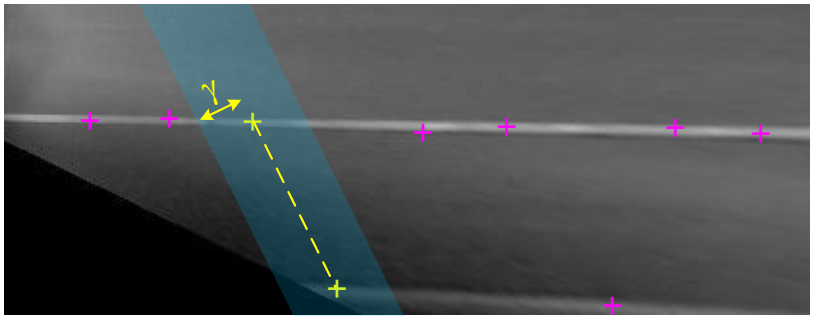
Figure 45: Inliers (red) and outliers (blue) present in H'_n.

outliers. Hence, these outliers must be removed. One way to do this is by applying RANSAC. RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers [23]. The pseudo-code for RANSAC is provided in Algorithm 2. RANSAC is performed by considering a set S that contains all the lane marker center points (both inliers and outliers) shown in Fig. 45. Then, we randomly select two points with replacement from S to create a line that is used as the model. Next, we determine the set of points S_p (subset of S) that are within a distance γ from the line. If the size of S_p (number of inliers) is more than T, then S_p is considered as the consensus set and its error is determined. The error is the maximum of the distance between the points in S_p and the line. This process is repeated $N = \binom{n}{k}$ times where n is the size of S and $k = 2$ since two points are required to describe a line. In each iteration, a random pair of points is selected

with replacement to describe the line and a new S_p is also created. Furthermore, if the size of the current iteration of $S_p > T$ and its error is less than the error of the consensus set, the consensus set is replaced by S_p and the new error is recorded as described on line 17 of Algorithm 2. In Fig. 46, we have shown two iterations of the



(a) Horizontal line model that contains all inliers.



(b) Diagonal line model that contains almost no inliers.

Figure 46: Two iterations of line models used in RANSAC.

RANSAC process where two different point pairs are used to create the line models. In Fig. 46a, the error will be small because most of the points will be within the distance γ from the line. On the other hand, in Fig. 46b, most points are greater than the distance γ from the line. Upon completing N trials, RANSAC will provide a consensus set that contains only the inliers [23] as shown in Fig. 47.

Once the outliers have been eliminated, we can proceed with curve fitting. Curve fitting is important to produce a smooth curve that represents the lane boundary in H'_n. Without curve fitting, the lane boundary curve will more than likely have a jagged shape due to the slight perturbations that may exist within the inliers. Although there are numerous way to perform this task, curve fitting is carried out

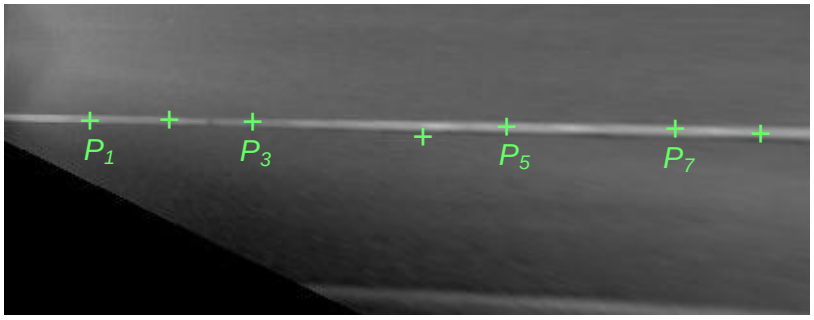


Figure 47: H'_n with inliers.

Algorithm 2 RANSAC algorithm [23]

$ConsensusSet = \{\}$
$S =$ Set of all points in Fig. 45
$error = L$ // *Very large number*
for N trials **do**
  $l =$ Line model created from two randomly selected points
  $S_p = \{\}$
  $err = 0$
  for Each Point P_i in S **do**
    $d =$ Distance between P_i and l
    if $d < \gamma$ **then**
      Add P_i to S_p
      if $d > err$ **then**
        $err = d$
      end if
    end if
  end for
  if $size(S_p) > T$ & $err < error$ **then**
    $ConsensusSet = S_p$
    $error = err$
  end if
end for

here using the Linear Least Squares (LLS) approach. LLS is a procedure for fitting a mathematical model to a given set of data points by minimizing the sum of squared errors between the data values and the modeled values [32]. To perform LLS, we describe a linear model of the form

$$\beta_1 x + \beta_2 = y \tag{10}$$

Therefore, each inlier in the consensus set can be described as

$$\beta_1 x_i + \beta_2 = y_i + e_i \tag{11}$$

Where x_i is the x co-ordinate of the point P_i, y_i is its correspond y co-ordinate, e_i is the error, and β_i are the unknown parameters that need to be determined. This can be rewritten in matrix form as

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{y} + \mathbf{e} \tag{12}$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_k & 1 \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}, \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_k \end{bmatrix} \tag{13}$$

and $\mathbf{e}$ is the error to minimize in the least squares sense. The least squares solution for $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{14}$$

In Eq. (10), we are creating a straight line model to represent the lane boundary. The linear equation can be modified to a quadratic by including x_i^2 and another parameter β_3, thus producing a quadratic curve to represent the lane boundary. Although, a curve is a better representation of the lane boundary, its model is easily affected by minor perturbation that may exist within the inliers. An example of such a situation is shown in Fig. 48 where a minor perturbation in the middle affects the estimation of the curve. In contrast, a straight line is more rigid, but its orientation is less affected by these minor perturbations. Finally, in Fig. 49, we have fitted a straight line model to the inliers in H_n' that is described by its slope-intercept (m, b) parameters.

2.4.7 Tracking

Output of the previous block is the set of slope-intercept parameters (m, b) that describe a line in the image that mostly represents the lane boundary. However,

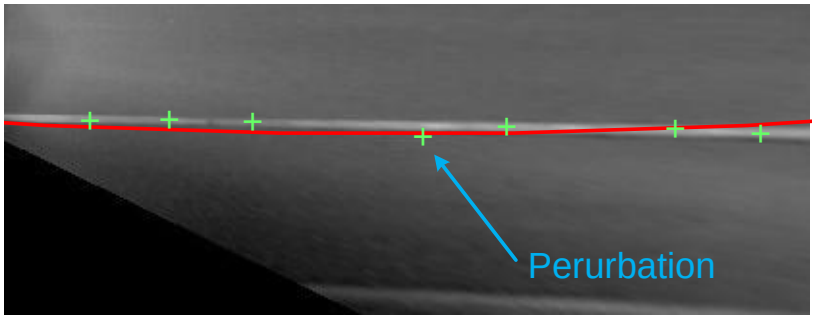


Figure 48: Minor perturbation in the points can affect the shape of the quadratic curve.

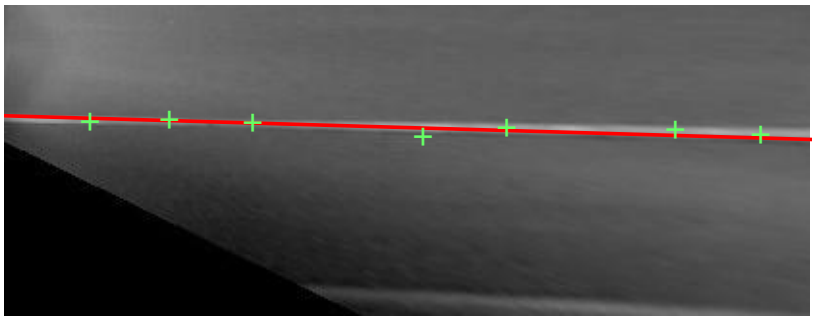


Figure 49: Minor perturbation in the points can affect the shape of the quadratic curve.

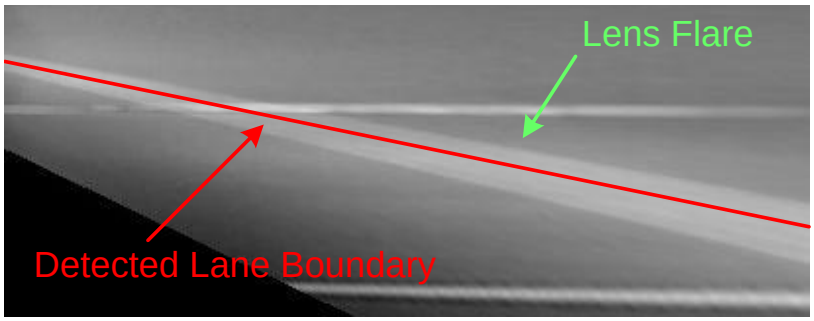
simply considering these parameters as the best descriptors for the lane boundary may not produce ideal results. This is because the position of the lane markers may move sporadically from one image to the next due to noise. The noise mentioned here can be caused by two phenomena:

1. False positives in the image that appear as lane markers and affect the detection process as shown in Fig. 50 (lens flares).

2. Bumps in the road that temporarily affect the camera's calibration parameters causing the lane markers to appear spatially shifted or skewed in the world image H'_n.

Let us take a closer look at phenomenon #2. In Fig. 51, we have provided three sequential images from a video clip. In Fig. 51b, the vehicle encounters a bump on the road that causes the lane maker in H'_n to appear spatially shifted from the

(a) Lens flare in the camera image.

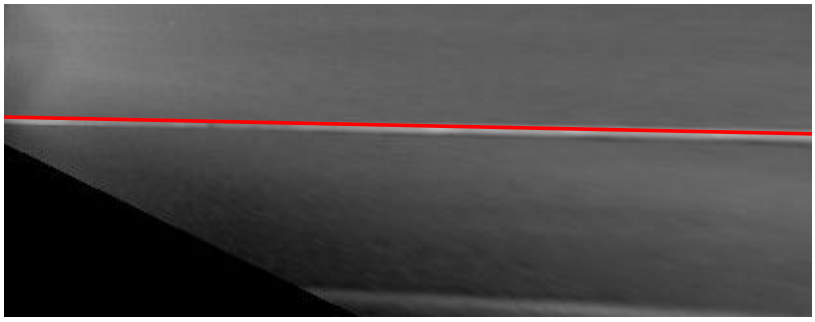


(b) Lens flare detected as a lane marker in the world image.

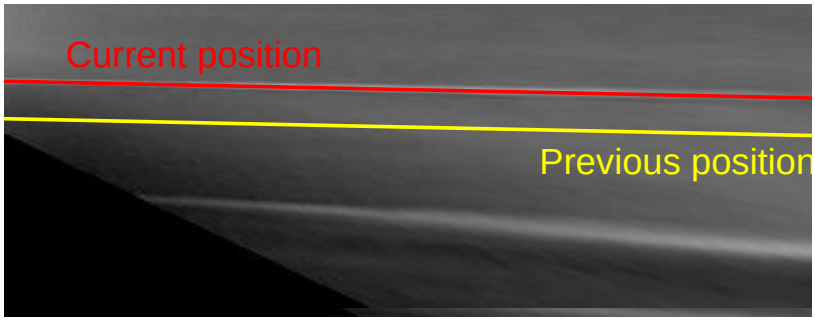
Figure 50: Lane marker detection affected by lens flare.

previous image. If the (m, b) parameters detected in each image are used as is, then, the line representing the lane boundary will move rapidly between the three images. Although the lane boundary is well represented in all the three images, the lane markers do not move in such a manner on the road; hence, the rapid movement is attributed by noise and needs to be attenuated. This is where tracking comes in.

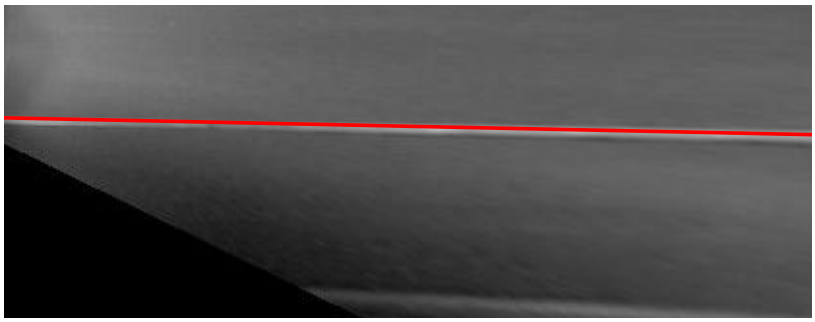
Tracking is used to estimate the position of the line representing the lane boundary from one image to the next in the video clip. Tracking is an important component of the ALD 1.0 because it prevents the line representation from changing sporadically or flickering due to the noise that may be present in the image. Although numerous different methods exist and can be used to perform tracking, we use the Kalman filter



(a) Lane boundary detected in the image.



(b) Bump on the road affects camera parameters and causes the position of the lane boundary to change its position.



(c) Lane boundary returns to the position in the first image.

Figure 51: Lane boundary moves rapidly between images.

because it is the most popular tracker used in lane detectors. The Kalman filter is an optimal linear estimator that recursively estimates the state of a linear dynamic system from a series of noisy measurements [33]. Since we are simply applying the Kalman filter for tracking, this section will only cover the details for setting up the parameters and matrices that enable its operation. Additional details regarding the theory and derivations of the Kalman filter can be found in [33].

With the Kalman filter, the idea is to track or estimate the parameters that describe the line representing the lane boundary from one image to the next. To do this, we first convert the line parameters provided by the data modeling block from the slope-intercept representation (m, b) to the polar representation (ρ, θ) where ρ is the perpendicular distance between the line and origin, and θ is the angle formed by the perpendicular and the origin. The polar representation was described previously in Sec. 2.4.4. This conversion is performed because in a few experiments, we observed that the small changes in line's position or orientation caused large changes in m or b which was many times problematic to the tracker. Therefore, the Kalman filter tracks ρ and θ using the measurements received from the data modeling block. Hence, the state vector is written as

$$\mathbf{x}_n = \begin{bmatrix} \rho_n & \dot{\rho_n} & \theta_n & \dot{\theta_n} \end{bmatrix}^T \tag{15}$$

Where ρ_n and θ_n are the parameters of the line and $\dot{\rho_n}$ and $\dot{\theta_n}$ are the derivatives of ρ_n and θ_n, respectively with respect to time. Considering the constant velocity model, the state equation

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{w}_n \tag{16}$$

can be rewritten as

$$\begin{bmatrix} \rho_{n+1} \\ \dot{\rho}_{n+1} \\ \theta_{n+1} \\ \dot{\theta}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho_n \\ \dot{\rho_n} \\ \theta_n \\ \dot{\theta_n} \end{bmatrix} + \mathbf{w}_n \tag{17}$$

where $\Delta T = 33$ ms (for 30 fps) is the time interval between each image in the video clip. Additionally, the process noise vector $\mathbf{w}_n \sim N(0, \mathbf{Q})$ is modeled in terms of

acceleration noises [34]. As a result, $\mathbf{Q}$, the covariance of the process noise is

$$\mathbf{Q} = \begin{bmatrix} \ddot{\sigma}_\rho^2 \cdot \frac{\Delta T^4}{4} & \ddot{\sigma}_\rho^2 \cdot \frac{\Delta T^3}{2} & 0 & 0 \\ \ddot{\sigma}_\rho^2 \cdot \frac{\Delta T^3}{2} & \ddot{\sigma}_\rho^2 \cdot \Delta T^2 & 0 & 0 \\ 0 & 0 & \ddot{\sigma}_\theta^2 \cdot \frac{\Delta T^4}{4} & \ddot{\sigma}_\theta^2 \cdot \frac{\Delta T^3}{2} \\ 0 & 0 & \ddot{\sigma}_\theta^2 \cdot \frac{\Delta T^3}{2} & \ddot{\sigma}_\theta^2 \cdot \Delta T^2 \end{bmatrix} \tag{18}$$

where $\ddot{\sigma}_\rho^2$ and $\ddot{\sigma}_\theta^2$ are the variances of the process noises associated with the accelerations in ρ and θ, respectively. Although $\mathbf{Q}$ is written in terms of $\ddot{\sigma}_\rho^2$ and $\ddot{\sigma}_\theta^2$, we were not able to determine these values from simple experiments. However, we were able to estimate σ_ρ^2 and σ_θ^2, the variances of the process noises associated with ρ and θ, respectively as explained below. The relationship between $\ddot{\sigma}_\rho^2$ and σ_ρ^2 is given as

$$\sigma_\rho^2 = \ddot{\sigma}_\rho^2 \cdot \frac{\Delta T^4}{4} \tag{19}$$

or

$$\ddot{\sigma}_\rho^2 = \sigma_\rho^2 \cdot \frac{4}{\Delta T^4} \tag{20}$$

A similar relationship holds for σ_θ^2. Hence, $\mathbf{Q}$ is rewritten in terms of σ_ρ^2 and σ_θ^2 as

$$\mathbf{Q} = \begin{bmatrix} \sigma_\rho^2 & \sigma_\rho^2 \cdot \frac{2}{\Delta T} & 0 & 0 \\ \sigma_\rho^2 \cdot \frac{2}{\Delta T} & \sigma_\rho^2 \cdot \frac{4}{\Delta T^2} & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & \sigma_\theta^2 \cdot \frac{2}{\Delta T} \\ 0 & 0 & \sigma_\theta^2 \cdot \frac{2}{\Delta T} & \sigma_\theta^2 \cdot \frac{4}{\Delta T^2} \end{bmatrix} \tag{21}$$

The variance σ_ρ^2 was determined experimentally as follows. We took three sequential images from a video clip that were indexed as n, $n+1$, and $n+2$. Each image contains ground truth (see Chap. 4 for details on ground truth). Then, for each image, the ground truth curve was transformed to world co-ordinates (transformation is described in Sec. 2.4.8), then, modeled as a straight line and described in terms of (ρ, θ) parameters. Thus we have three pairs (ρ_n, θ_n), $(\rho_{n+1}, \theta_{n+1})$, and $(\rho_{n+2}, \theta_{n+2})$ for the three images that refer to the ground truth. Then, $\dot{\rho}_{n+1}$, the derivative of ρ

with respect to time in the image $n+1$, is estimated as

$$\dot{\rho}_{n+1} = \frac{\rho_{n+1} - \rho_n}{\Delta T} \tag{22}$$

then,

$$\hat{\rho}_{n+2} = \rho_{n+1} + \dot{\rho}_{n+1} \cdot \Delta T \tag{23}$$

and the error is computed as

$$\text{error} = \rho_{n+2} - \hat{\rho}_{n+2} \tag{24}$$

This error computation is similarly performed on 100 different image triplets and used to create a distribution. Finally, assuming that the error has a zero mean Gaussian distribution [33], the standard deviation of the distribution is computed to give $\sigma_\rho = 0.56$. Similarly, $\sigma_\theta = 0.211$ is computed as the standard deviation on the error distribution of θ.

The measurement equation for the Kalman filter is

$$\mathbf{z}_n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_n + \mathbf{v}_n \tag{25}$$

where the measurement noise vector $\mathbf{v}_n \sim N(0, \mathbf{R})$ and $\mathbf{R}$, the covariance of the measurement noise is

$$\mathbf{R} = \begin{bmatrix} \sigma_{v,\rho}^2 & 0 \\ 0 & \sigma_{v,\theta}^2 \end{bmatrix} \tag{26}$$

where $\sigma_{v,\rho}^2$ and $\sigma_{v,\theta}^2$ are variances of the measurement noise associated with ρ and θ, respectively. The variance $\sigma_{v,\rho}^2$ was determined experimentally as follows. We took 100 images that contained ground truth and described the lane boundary curve using (ρ, θ) parameters as mentioned above. Thus we have 100 pairs (ρ_1, θ_1), (ρ_2, θ_2) ... for the 100 images. Then, we performed lane detection on each of the 100 images and model each detected lane boundaries in terms of $(\hat{\rho}, \hat{\theta})$ as well ($\hat{\rho}$ because these are

the measurements); therefore, we have $(\hat{\rho}_1, \hat{\theta}_1)$, $(\hat{\rho}_2, \hat{\theta}_2)$ ... for the 100 images. The error is computed as

$$\text{error} = \rho_1 - \hat{\rho}_1 \tag{27}$$

and a distribution is created from the collection of error values. Again, assuming that the error has a zero mean Gaussian distribution [33], the standard deviation of the distribution is computed to give $\sigma_{v,\rho} = 0.89$. Using a similar procedure, $\sigma_{v,\theta} = 1.01$ is also determined. This concludes setting up the parameters and the matrices for the Kalman filter.

When the Kalman filter receives its first set of measurements that correspond to the lane boundary in H_1' (subscript $n = 1$ corresponds to image index described in Sec. 2.4.1), the initial values of the parameters in the state vector are not perfectly know; hence, the error covariance matrix $P_{1|1}$ is initialized with a large number L on the diagonal. As a result, the Kalman filter will then prefer to use the first set of measured values over the values in the model. However, this preference will change over the remaining images in the clip.

Let us consider an image H_c' (some random image in the video clip) in which no lane boundary is detected. This situation could arise under two circumstances:

1. The lane markers on the road are worn out making them difficult to detect in H_c'.

2. There are no lane markers present in H_c'.

In either circumstance, the feature extraction block will provide no lane marker location information. As result, the data modeling block will be unable to estimate a lane boundary curve and therefore will not provide any output or measurements to the Kalman filter to base its estimates. Hence, in this condition, the Kalman gain K (not the same parameter from Eq. (1)) is set to zero forcing the Kalman filter to rely

purely on its prediction to produce the estimate as shown below

$$\hat{\mathbf{x}}_{n|n-1} = \hat{\mathbf{x}}_{n|n} \tag{28}$$

until a measurement arrives. However, the Kalman filter is only allowed to operate purely on its prediction for a maximum of 60 sequential images or 2 seconds. After 60 images, the filter is disabled and no estimates are produced. This prevents the filter from "drifting" and producing incorrect estimates as no lane markers may be present in H'_{c+61}. If the filter has been disabled and a measurement is received; then, the Kalman filter is reset and the error covariance matrix is initialized with a large value L along its diagonal.

Finally, the line parameters (ρ, θ) that are estimated by the Kalman filter are used to plot a line that represents the lane boundary in H'_n. However, as per the requirements discussed in Sec. 2.2, the lane boundary must be presented in the camera image. Therefore, the line undergoes a perspective mapping (inverse of the IPM) to show its representation of the lane boundary in $I(n)$. This mapping is illustrated in Fig. 52. The perspective mapping is computed by plugging in the (x, y) co-ordinates of each point on the line into Eq. (3)−(5) to retrieve its corresponding (r, c) co-ordinates in $I(n)$.

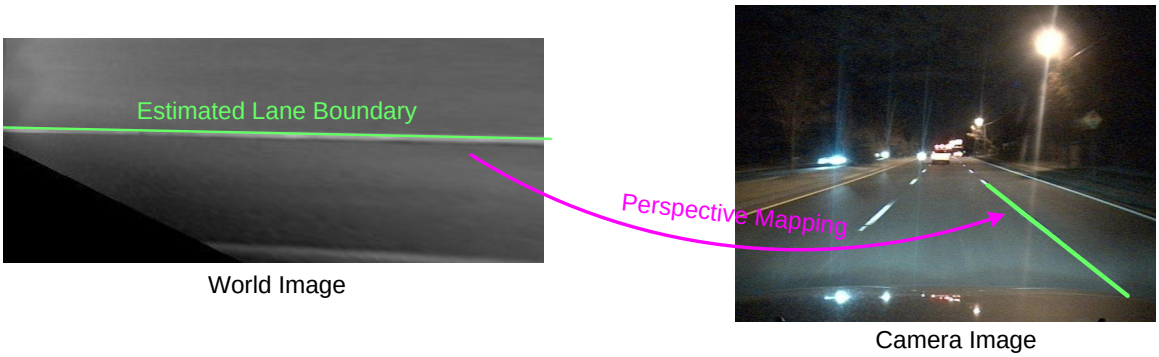


Figure 52: Estimated lane boundary mapped from the H'_n to the camera image $I(n)$.

Tracking using the Kalman filter helps produce estimates that are close to the ground truth as will be shown in Sec. 2.4.9. As previously mentioned, the Kalman

filter that we use assumes the constant velocity model. The Kalman filter can be generalized by incorporating acceleration or higher derivatives into the state vector. Additionally, the use of a particle filter based tracker could perhaps improve the results. However, the constant velocity model used by the Kalman filter described in this section not only produces good results that are sufficient for this work, it appears to be the model most commonly implemented by Kalman filter based trackers that are used for lane detection.

This block completes the estimation process for the right lane boundary. The two feature extraction steps, data modeling and tracking are repeated on the top half of the binary image B_n and the world image W_n to estimate the left lane boundary in $I(n)$ [24].

2.4.8 Error Calculation

To determine the accuracy of the ALD 1.0, the estimated lane boundary needs to be compared to the ground truth. Since both ground truth and the estimated lane boundary are curves, this comparison is done by measuring distances between the estimate and the ground truth at a number of locations along the length of the ground truth as shown in Fig. 53. However, these distances are not measured directly

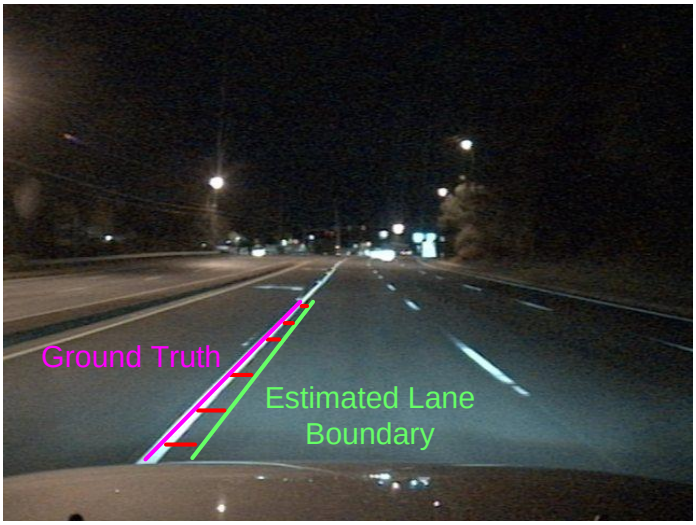


Figure 53: Distance between the estimate and ground truth at various locations.

in the camera image. This is because the distance corresponding to the gap between two pixels in the real-world changes depending on the location of the pixel pair in the camera image. To illustrate, consider two line segments in Fig. 54. Both segments are

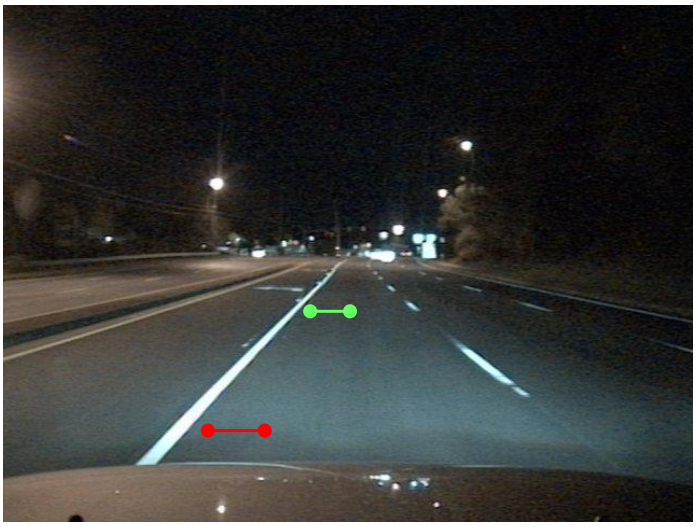


Figure 54: Line segments corresponding to different lengths.

15 pixels in length, but the red line corresponds to a distance of 5 ft in the real-world. Whereas the green line corresponds to a distance of 20 ft. The non uniform distance mapping is caused by the perspective effect and needs to be nullified. Therefore, both the ground truth and the estimated lane boundary in the camera image are mapped to the world co-ordinates. This mapping is similar to the IPM transformation provided in Sec. 2.4.2 except here the (r, c) co-ordinates of each curve in the camera image are mapped to (x, y) co-ordinates in the world image. This mapping is illustrated in Fig. 55 and provided below

$$x'(r) = h \cdot \left(\frac{1 + \left[1 - 2 \cdot \left(\frac{r-1}{M-1}\right)\right] \tan\alpha_v \tan\theta_o}{\tan\theta_o - \left[1 - 2 \cdot \left(\frac{r-1}{M-1}\right)\right] \tan\alpha_v} \right) \tag{29}$$

$$y'(r, c) = h \cdot \left(\frac{\left[1 - 2 \cdot \left(\frac{c-1}{N-1}\right)\right] \tan\alpha_u}{\sin\theta_o - \left[1 - 2 \cdot \left(\frac{r-1}{M-1}\right)\right] \tan\alpha_v \cos\theta_o} \right) \tag{30}$$

where

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{31}$$

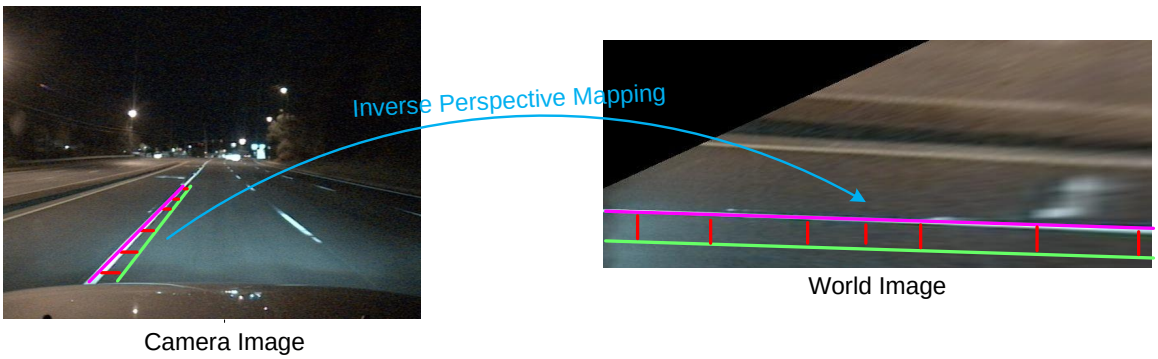


Figure 55: Mapping the ground truth and lane boundary estimate to world coordinates.

Once the mapping is complete, measurements are performed by computing the distance between the ground truth curve and the estimated lane boundary curve at specific locations in the world image [35]. These locations are indexed by i where $i \in [11, 30]$ ft in the world image and are spaced one foot apart (every 8 pixels). A λ measurement is computed as

$$\lambda_{(i,n)} = \max\left(|Gt_{(i,n)} - X_{(i,n)}| - \frac{T}{2}, 0\right) \tag{32}$$

where $Gt_{(i,n)}$ is the location of the ground truth and $X_{(i,n)}$ is the location of the estimated lane boundary at i ft in the world image W_n and in image n of the video clip. The computation of the measurements is illustrated in Fig. 56 where the green line represents the estimated lane boundary and the purple line is the ground truth. It can be seen that λ is computed at several locations along the length of the ground truth. T is used as an interval around the ground truth at each location i in Fig. 56 and set to 0.5 ft. The reason for using this interval is because a normal lane marker has a thickness of approximately 6 inches or 0.5 ft as stated by the (FHA) [19]. Therefore,

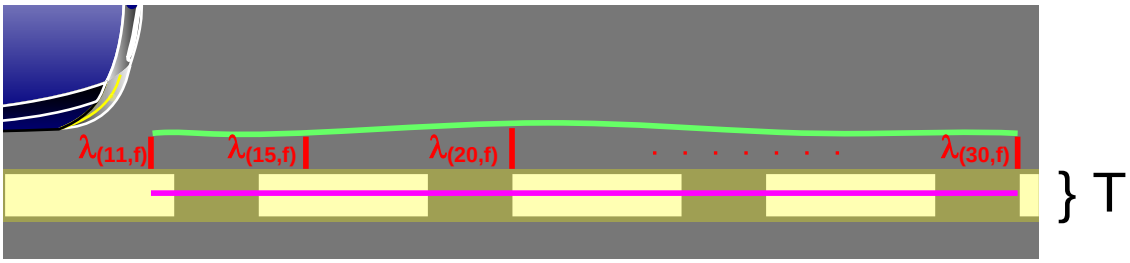


Figure 56: Calculating errors using λ offsets.

if a lane boundary estimate falls within the interval at i, then it is categorized as having no error. Finally, the error in each image n of the video clip is computed as

$$\mathrm{E(n)} = \|\boldsymbol{\lambda}\|_\infty = \max_i \lambda_{(i,n)} \tag{33}$$

where E(n) is the L_∞ norm of the λ values and its unit is ft.

2.4.9 Performance Evaluation

Now that we described the inner workings of the ALD 1.0, its performance must be evaluated. Performance evaluation is an important part in the development of the ALD 1.0 because it involves testing the lane detector with representative data and quantifying its output with regard to the ground truth that is subject to certain performance measures. Five performance measures are used in the evaluation and they are described as follows:

1. Correct Detection (CO) occurs when the ground truth exists and a lane boundary is detected in the image. In addition, at most $\frac{M}{2}$ of the error measurements, λ, from Sec. 2.4.8 are greater than 0. This implies that the estimated lane boundary curve is very close to the ground truth curve and possibly suffers minor alignment issues. An illustration of this performance measure is shown in Fig. 57. $M = 20$ since 20 λ measurements are performed along the length of the ground truth curve.

2. Misalignment (MA) occurs when the ground truth exists and a lane boundary is detected in the image. In addition, more than $\frac{M}{2}$ of the error measurements,

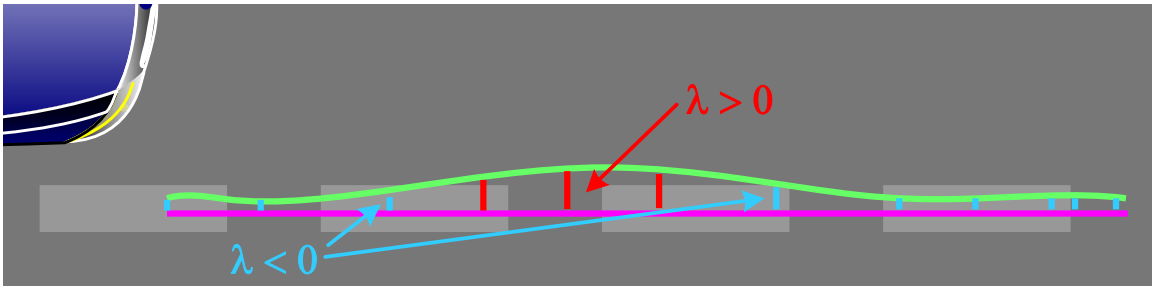


Figure 57: Lane boundary estimate with minor alignment issues.

λ, are greater than 0. An illustration of this performance measure is shown in Fig. 58. This implies that the estimated lane boundary curve is noticeably misaligned with respect to the ground truth.

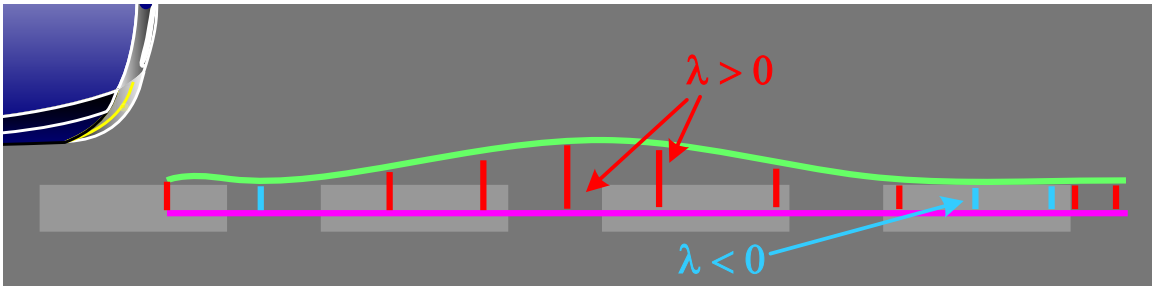


Figure 58: Lane boundary estimate with noticeable alignment issues.

3. Missed Detection (MD) occurs when the ground truth exists in the image but no lane boundary has been detected.

4. False Detection (FD) occurs when a lane boundary has been detected but no ground truth exists in the image.

5. True Rejection (TR) occurs when no ground truth exists in the image and no lane boundary has been detected.

The testing data consists of eight video clips; four clips contain day time driving footage and the other four contain twilight or night time driving footage. Each clip is approximately 1 minute in duration ($\sim$ 1800 images) and features variations in road surface quality, traffic, illumination, and shadow conditions to include a realistic

set of conditions that one would encounter while driving on city roads and highways. Moreover, these clips were acquired from our database which is detailed in Chap. 5 of this dissertation.

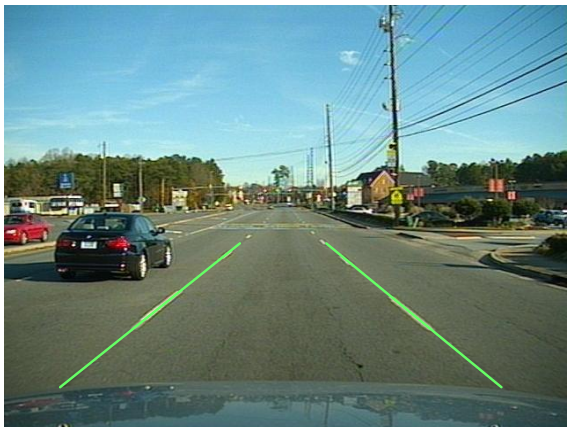
If we refer back to Object of Interest Detection in Sec. 2.4.3, the two parameters α, the threshold for local intensity variation, and k, the height of the window were determined empirically. Therefore, testing was performed on each of the eight video clips using a combination of different values for both parameters, where $\alpha \in [7, 12, 17, 22, 27]$ and $k \in [3, 5, 7, 9, 11]$. The pair of values that yielded the most number of correct detections (CO) among the eight clips was considered to be the best pair and was used in the ALD 1.0. The performance of the ALD 1.0 using the best pair of α and k is reported in Table 2. We have also provided the performance of the ALD 1.0 for all combinations of α and k that were used in testing in Appendix A. For verification, the total of performance measures CO + MA + MD = 100% and FD + TR = 100% for each clip. However, in the situations that no false detections are encountered, then FD = TR = 0. The performance measures for the left and right lane boundaries are averaged and reported in Table. 2. Furthermore, $\sigma_{\mathrm{E(n)}}$ is the standard deviation of the distribution of the error E(n) in the clip. Although E(n) is in units of ft as mentioned in Sec. 2.4.8, it is converted to inches for illustration and is reported in Table. 2.

In Fig. 59, we have provided images that illustrate the lane boundaries estimated by the ALD 1.0 as green lines. The lane detector is able to detect the lane boundaries on city roads and highways, and also in the presence of traffic and changing illumination. The error E(n) was only computed for the images in which a correct detection occurred. Furthermore, the histogram of E(n) for each clip is also provided in Appendix A. Although the False Detection (FD) values are alarmingly high in Table 2, it is not of much concern because only a handful of images in the video clip lacked ground truth. For example, while crossing an intersection where no lane

Table 2: Performance of the ALD 1.0 on the 8 video clips using $\alpha = 7$ and $k = 5$. The number of images in which false detections and true rejections were encountered are provided in the parenthesis of their columns.

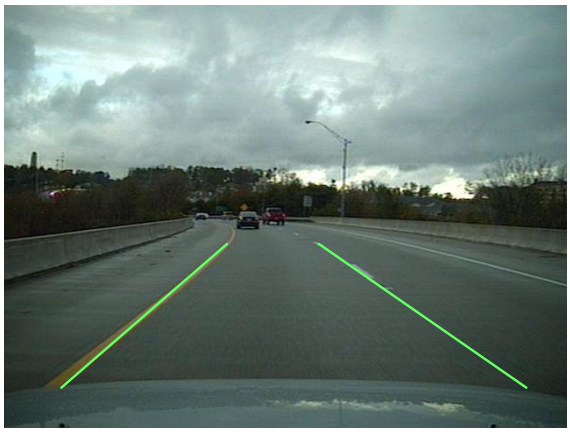
Name	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
Clip 1	47.43	51.53	1.04	100.00 (61)	0.00 (0)	1.93
Clip 2	83.38	15.34	1.28	0.00 (0)	0.00 (0)	0.90
Clip 3	88.21	10.42	1.37	0.00 (0)	0.00 (0)	0.84
Clip 4	76.62	21.72	1.66	0.00 (0)	0.00 (0)	0.56
Clip 5	90.19	7.17	2.64	0.00 (0)	0.00 (0)	0.43
Clip 6	82.65	15.31	2.04	0.00 (0)	0.00 (0)	0.58
Clip 7	96.74	1.68	1.59	0.00 (0)	0.00 (0)	0.80
Clip 8	71.00	27.59	1.42	100.00 (65)	0.00 (0)	0.49

markers are present, the ALD 1.0 assumes that the lane markers are worn out and hence undetectable, therefore, the Kalman filter relies only its prediction to produce the estimate as explained in Sec. 2.4.7 leading to false detections. As a result, we have provided the number of images in the parenthesis of the FD column to give a better idea of the number of false detections that occurred in the clip. We have also provided the number of images in which true rejections occurred in the parenthesis of the TR column. In Fig. 60 and 61, we have shown a comparison between the ground truth, the measurement, and the Kalman filter estimates for ρ and θ on three road conditions. The measurement and the Kalman estimates showed some error during certain portions of a lane change; however, the Kalman estimated ρ and θ were very close to the ground truth at most times as shown in the plots. One point of concern was the potential change in the camera's pitch that would occur during driving. However, the camera's pitch was not affected by normal vibrations encountered while driving. The few times that the camera pitch was modified was when a small bump was encountered or at a point where there was a significant change in surface inclination. In either case, the modification was temporary and only in a few situations did it affect the detection of the lane markers, e.g. in Fig. 62d. However, the Kalman filter helped in smoothing out these bumps [26].



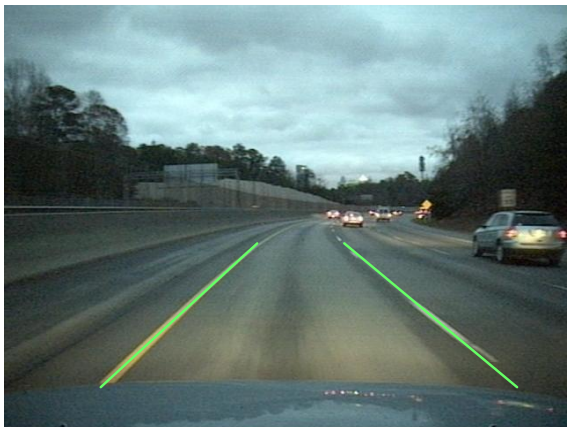
(a) City road.

(b) Highway with traffic.



(c) Curving highway without traffic.

(d) City road.

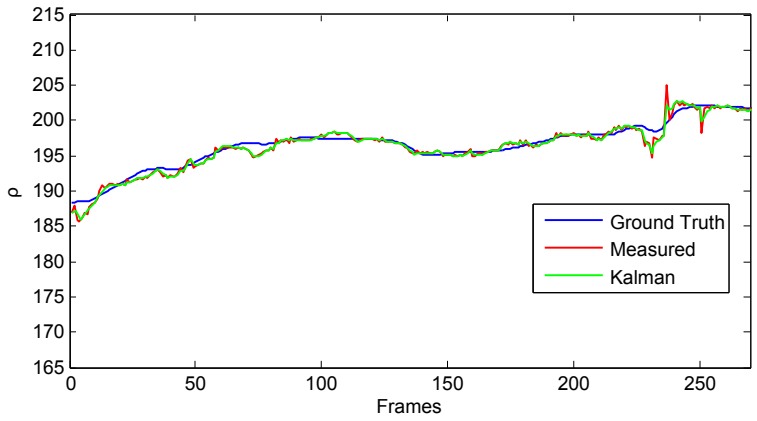


(e) Highway.

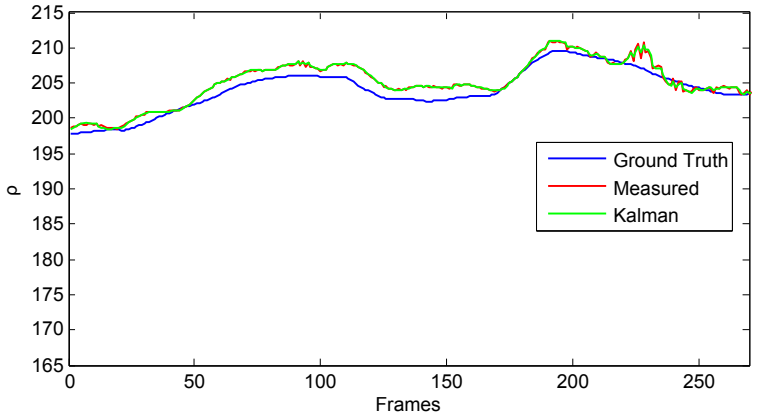
(f) Highway during a lane change.

Figure 59: Examples of correct lane detection.

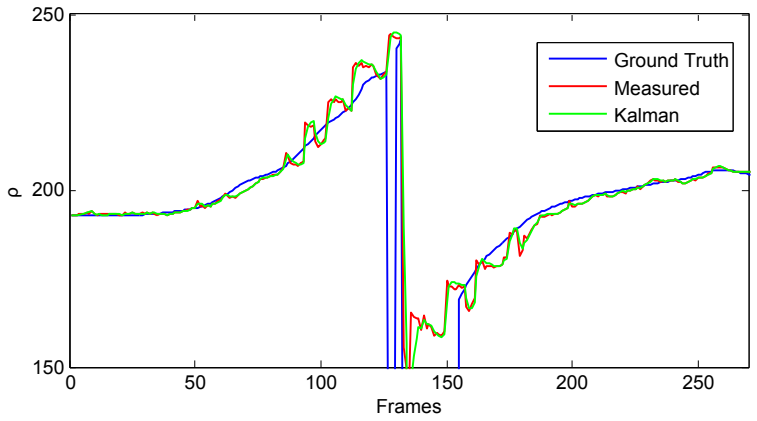
A few instances of missed detections (MD) and misalignments (MA) are shown in Fig. 62. The most common cause of these two detections was the absence of lane



(a) Straight road

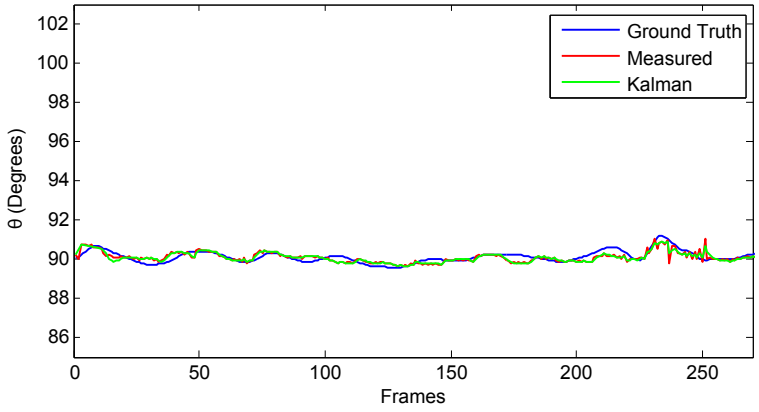


(b) Curving road

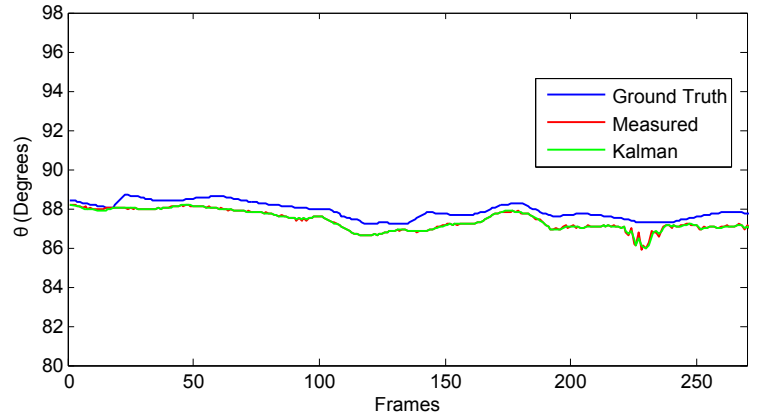


(c) Lane change

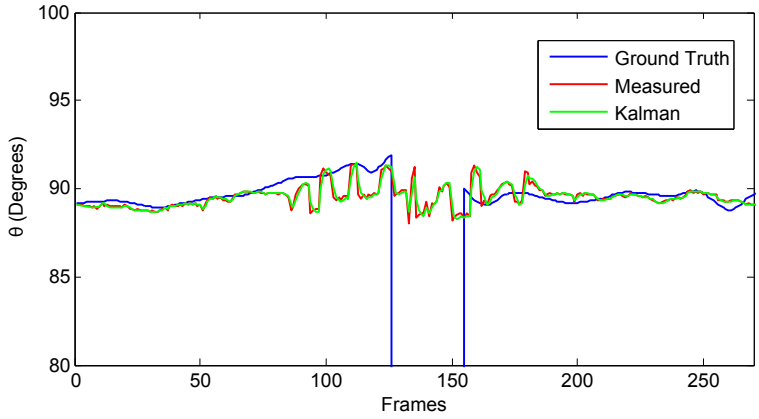
Figure 60: Comparison between the ρ values on a straight road, a curving road, and during a lane change. The blue line represents the ground truth, the red line represents the measured values, the green line represents estimates determined by the Kalman filter.



(a) Straight road



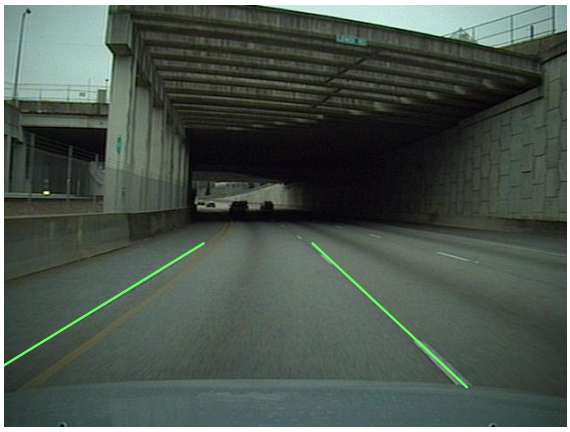
(b) Curving road



(c) Lane change

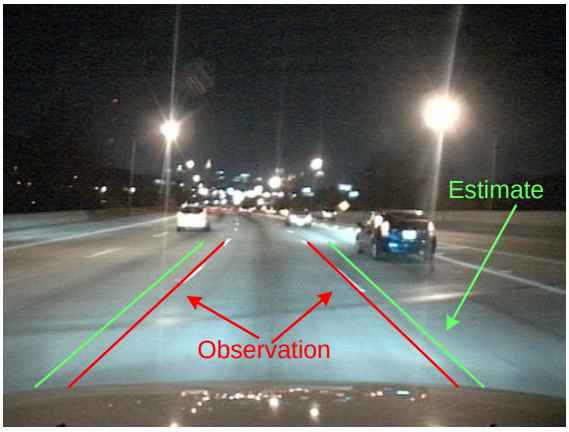
Figure 61: Comparison between the θ values on a straight road, a curving road, and during a lane change. The blue line represents the ground truth, the red line represents the measured values, the green line represents estimates determined by the Kalman filter.

(a) Worn out lane marker on the left causes missed detection.



(b) Crack next to the road is detected instead of the lane marker.

(c) Step ramp of a truck is detected.

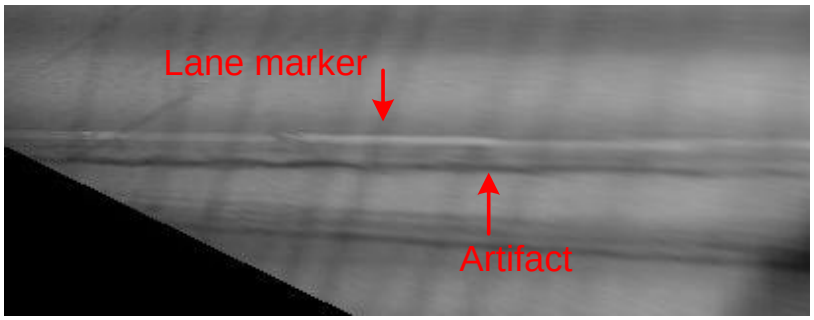


(d) Bump in the road causes sudden change in lane boundary position.

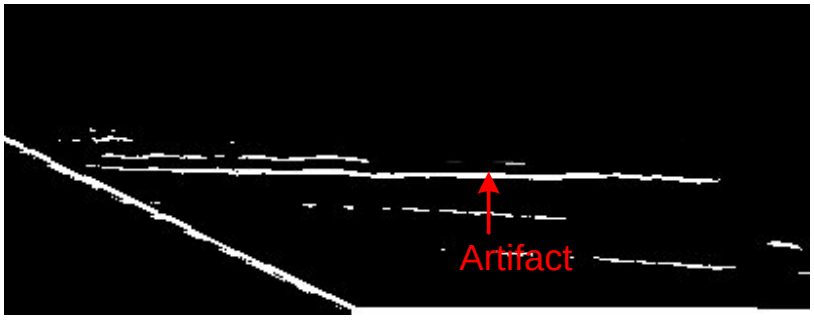
Figure 62: Examples of missed detections and misalignments.

markers often due to age and wear on local city roads. This occasionally leads to the detection and tracking of false signals such as cracks, tar patches, lens flares, and parts of neighboring vehicles. We also observed that the ALD 1.0 was able to produce good rates for correct detections when it was tested on the clips that contained footage of driving at night. However, it faced some difficulty when tested with day time footage as shown in Fig. 62a – 62c. The low correction detection rate in Clip 1 stems from the Object of Interest block (details in Sec. 2.4.3) that has difficulty in differentiating between lane markers from other deformities on the road surface that are visible during the day. An example of this problem is illustrated in Fig. 63,

where the grayscale image H'_n is shown in Fig. 63a and the binary image H_n (from the Object of Interest block) is shown in Fig. 63b. In H_n, it can be observed that



(a) Lane marker is visible in H'_n.



(b) In the binary image H_n, the lane marker is not detected.

Figure 63: Problems detecting lane markers in the day.

the artifact is detected but not the lane marker. Consequently, the artifact may be estimated and tracked in future images of the video clip leading to a misalignment or missed detection.

2.4.10 Closing Remarks

In this section, we have introduced a new methodology for lane detection that is called the Advanced Lane Detector (ALD) 1.0. The methodology begins by preprocessing the captured images to highlight lane markers. Then, the images are geometrically transformed by application of Inverse Perspective Mapping (IPM) giving them the appearance of a bird's-eye view. Next, lane markers are detected in the transformed image with the help of an adaptive threshold, and then, using a series of template matching procedures, the exact locations of lane markers are determined. The outliers

among the lane markers are eliminated using RANSAC and, finally, the inliers are fitted with a curve that is tracked using a Kalman filter. The ALD 1.0 is implemented in Matlab and operates at about 2 fps on the test bed. The test bed is an Intel Core 2 [36] based computer running Windows XP [37] with 4GB of memory. With code optimization and implementation on dedicated hardware, a real-time system could be realized.

The ALD 1.0 primarily showed good results when tested with night time footage but faced some difficulty with day time footage as discussed in the previous section. This makes its application limited as a lane detector needs to be able to function in almost any illumination condition. Therefore, in the next section, we introduce a new lane detection methodology that is not restricted to any particular illumination condition.

2.5 Advanced Lane Detector 2.0

In this section, we present a new lane detection methodology that is capable of operating in almost any illumination condition. This new lane detection methodology is called the ALD 2.0 and its layout is shown in Fig. 64. First, images captured by the camera system undergo Inverse Perspective Mapping (IPM) [20] and a Color Transformation to aid in feature extraction. Next, in the Filtering block, the transformed images undergo Template matching followed by a Hysteresis threshold to create bi-

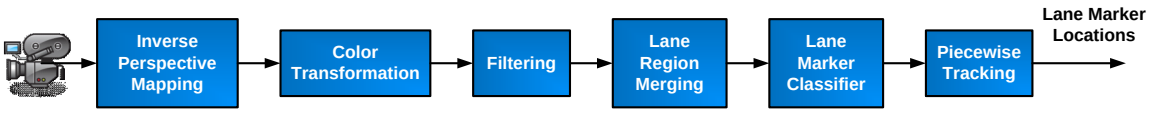


Figure 64: Layout of the ALD 2.0.

nary images. Then, Lane Region Merging block is used to combine these binary images into a single binary image in which the lane markers are detected. This is followed by a Lane Marker Classifier, which is used to ignore most of the artifacts that may be present in the binary image. Finally, the lane boundary is estimated by tracking the lane marker movement from one image to the next using a Piecewise Tracker. The ALD 2.0 is tested with videos that reflect real world driving conditions and its performance is discussed later in the section.

As in the ALD 1.0, it was our intention to include an improved Temporal Blurring approach (see Sec. 2.4.1 for details on Temporal Blurring) in the ALD 2.0 that will dynamically alter the parameters that are used to create the Average Image. For example, the number of images K as well as the offset $i \cdot \Delta$ can be modified depending on the speed of vehicle. Additionally, a spatial shift can be applied to the past images based on the vehicles movement to prevent lane markers from appearing thicker. However, the data set that we used only provided the speed of the vehicle. To compute the necessary spatial shift, we need to estimate the ego motion through

GPS co-ordinates and steering angle of the vehicle, which were unavailable at the time. Hence, we did not include the Temporal Blurring in the ALD 2.0.

Additionally, we observed that color in an image is useful in extracting lane markers from other objects on the road. Hence, the first two blocks in the ALD 2.0 shown in Fig. 64 operate on color images rather than grayscale images.

Now, we explain the details of each block of the new lane detector.

2.5.1 Inverse Perspective Mapping

Inverse Perspective Mapping (IPM) is a geometric transformation that is applied to the camera image to give it the appearance of a bird's-eye view [20]. Since IPM has been explained previously in this dissertation, please refer to Sec. 2.4.2 for more details. The camera image $I(n)$ and transformed world image W_n are shown in Fig. 65.

(a) Camera image $I(n)$. (b) Geometrically transformed world image W_n.

Figure 65: Inverse Perspective Mapping.

2.5.2 Color Transformation

The world image W_n from the IPM block undergoes a color transformation from its original RGB color space to YCbCr [35]. For each pixel (x, y) in W_n, the color

transformation from RGB to YCbCr [38] is given by

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{34}$$

where R, G, B are the red, green, and blue values of the pixel at (x, y) in the RGB colored image W_n while Y, Cb, and Cr are the computed luminance and chrominance components of the pixel (x, y) in color transformed image $\hat{W}_n$. We perform this transformation because in YCbCr, the luminance and chrominance components of a pixel are modeled separately. Hence, the effects of shadows and dynamic illumination in the chrominance components are reduced. This aids the detection of colored objects that are observed under a variety of illumination conditions on the road surface. In Fig. 66, we have shown the three color channels of W_n, while in Fig. 67, we have shown the three color channels of $\hat{W}_n$. It can be observed in Fig. 67, that white lane markers are prominently visible in the Y channel and yellow lane markers are prominently visible in both Cb and Cr channels. This distinction is particularly useful in the next block when we perform Filtering.

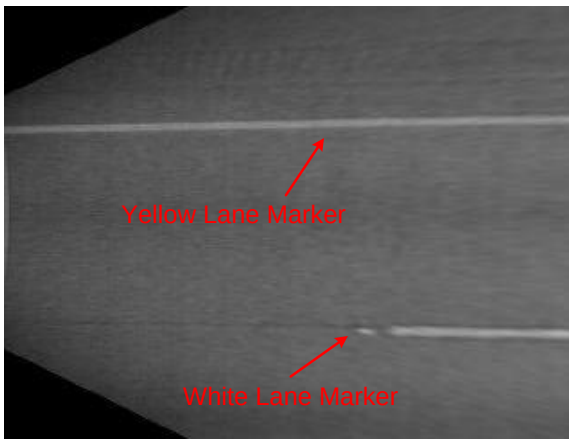
2.5.3 Filtering

This goal of this block is to detect lane markers in the image. As shown in Fig. 68, it consists of two stages:

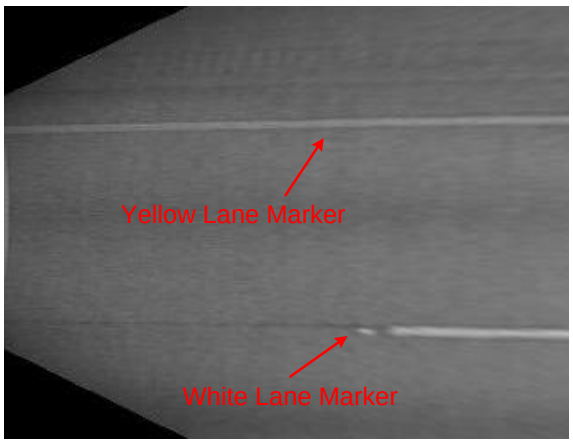
1. Template matching

2. Hysteresis threshold

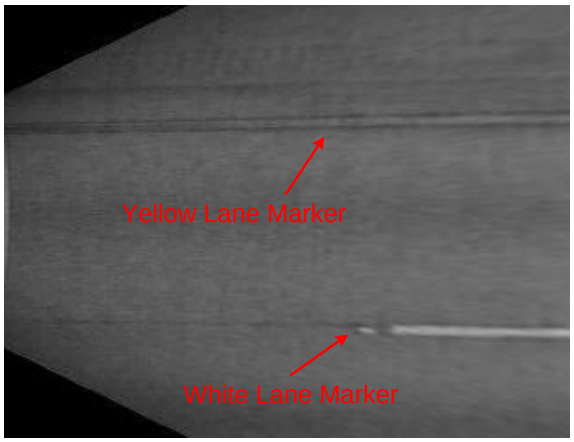
As in Sec. 2.4.5, Template matching is also performed by Normalized Cross Correlation (NCC) [30]. However, in the ALD 2.0, NCC is performed over the entire world image $\hat{W}_n$ rather than just in the Sampling Columns, which is the case with the ALD 1.0. Additionally, the templates used here are also in 2-D. Since normal,



(a) Red channel.



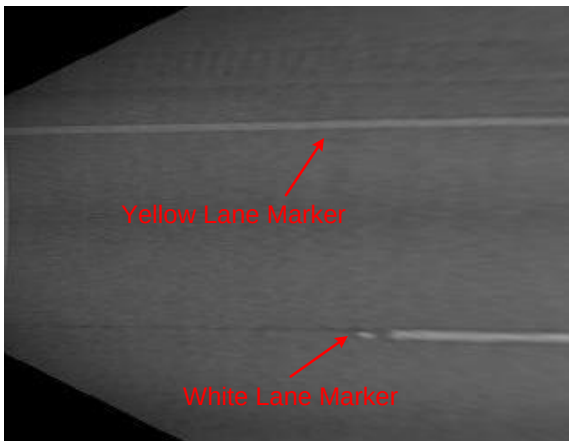
(b) Green channel.



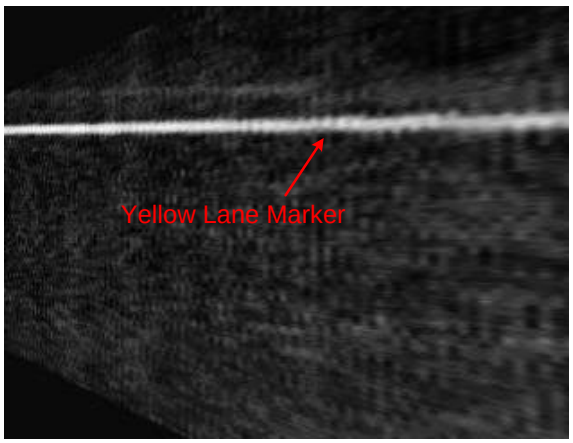
(c) Blue channel.

Figure 66: Individual channels of the RGB image with lane markers annotated.

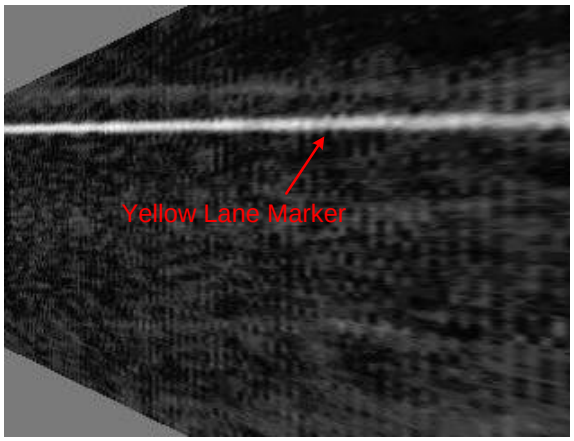
wide, and double lane markers are most commonly encountered on the road, we create templates for these three markers. The templates, with their dimensions are shown in Fig. 69. These templates are created to model the appearance of lane markers in $\hat{W}_n$. Since the shortest length of a lane marker is 1 ft [19], we set the lengths of the templates to 8 pixels which is the equivalent of 1 ft in the world image. In Fig. 70, we have shown the coefficients map that is computed as a result of performing NCC between the Y channel of $\hat{W}_n$ and the normal lane marker template. The coefficients map is a grayscale image with intensity values in the range $[-1, 1]$ for each pixel. In the coefficients map, areas similar to the template i.e. the lane markers, produce high coefficient value (values that are close to 1) as shown in Fig. 70. NCC



(a) Y channel.



(b) Cb channel (intensities are inverted in this image).



(c) Cr channel.

Figure 67: Individual channels of the YCbCr image with lane markers annotated.

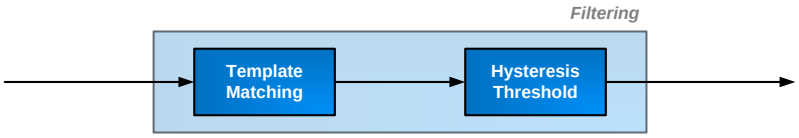


Figure 68: Stages in the filtering block.

performs a normalization of intensity values under the template [30]; hence, it can perform accurate template matching even under a variety of illumination conditions. Unfortunately, this also results in moderately high coefficient values for stray objects on the road surface which are prevalent in Fig. 70. Consequently, a threshold is applied to eliminate false detections. However, the use of a single threshold is not

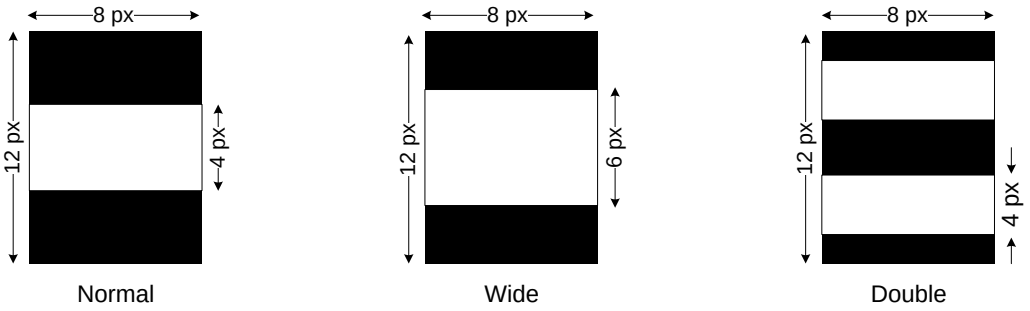


Figure 69: The three lane marker templates and their dimensions.

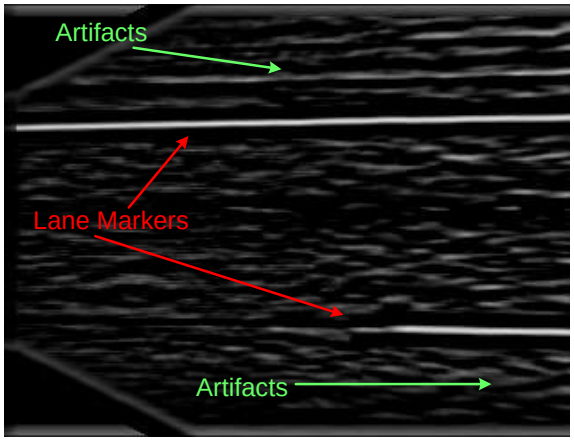
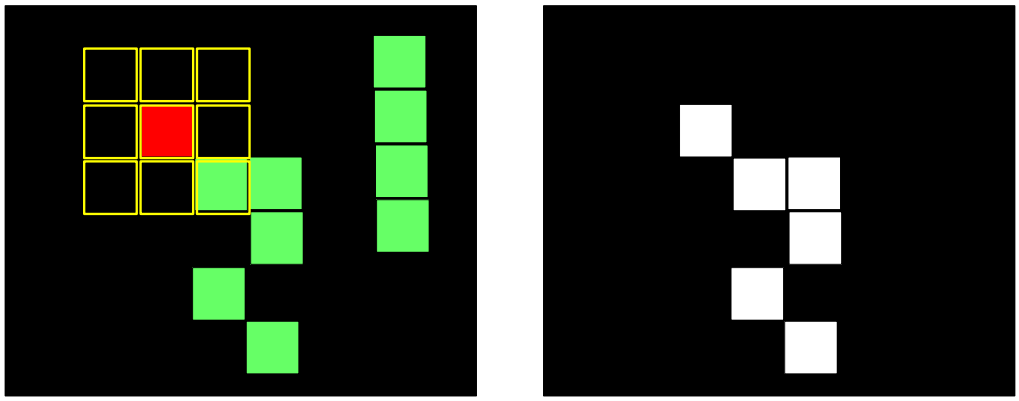


Figure 70: NCC coefficients map.

sufficient to accomplish this task. With a low threshold, noise will be an issue. On the other hand, with a high threshold, lane markers may become fragmented. Hence, the binary image is generated using a Hysteresis threshold [39].

In the Hysteresis threshold, a high threshold (τ_{Hi}) and a low threshold (τ_{Lo}) are specified. Any pixel with a coefficient value greater than τ_{Hi} is set to a binary one. Then, pixels within an 8-connectivity neighborhood of those that have been set equal to one are searched. If the value of any neighboring pixel is greater than τ_{Lo}, then it is also set to one. This process is illustrated in the synthetic image in Fig. 71. The τ_{Hi} pixels shown in red and τ_{Lo} pixels shown in green. Consequently, stand-alone low threshold pixels are ignored and a more "connected" result is produced by joining high and low threshold pixels. In Fig. 72, we have shown the coefficients map in Fig. 70 that is converted to binary after hysteresis threshold.



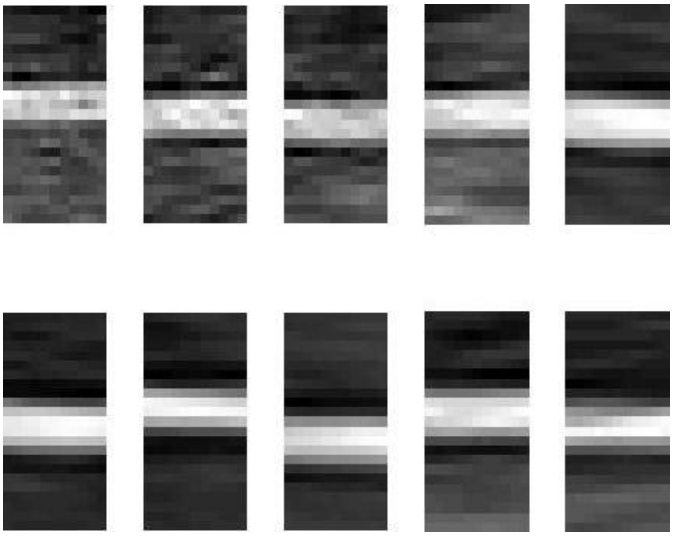
(a) Pixels exceeding the high (τ_{Hi}) and low (τ_{Lo}) thresholds are shown in the image in red and green, respectively. The yellow grid represents the 8-connectivity neighborhood around the red pixel.

(b) Resultant binary image. The stand-alone low threshold pixels are ignored.

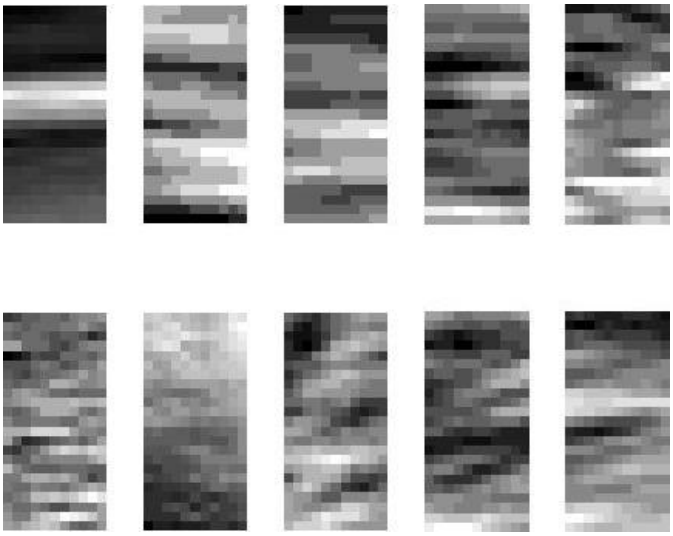
Figure 71: Illustration of hysteresis threshold.

Figure 72: Grayscale coefficients map converted to binary after hysteresis threshold.

To determine τ_{Hi}, we adopted the same approach from Sec. 2.4.5; however, instead of using intensity profiles as in the ALD 1.0, here we used small image patches that contain lane markers. We have shown a few sample patches in Fig. 73. As in Sec. 2.4.5, NCC is performed on each patch and the correlation coefficients are collected. These correlation coefficients serve as the positive training samples. For negative training samples, we took patches of the road surface from $\hat{W}_n$ that did not contain the lane markers and collected the correlation coefficients. In Fig. 74, we have presented the ROC (Receiver Operator Characteristics) [31] curve for the 160



(a) Positive training samples.



(b) Negative training samples.

Figure 73: Few training samples used to determine the threshold.

training instances. The cut off point on the ROC curve that produced the highest accuracy is set as the threshold τ_{Hi}. We have also provided a confusion matrix in Table 3, which contains the predicted outcomes for the 160 training instances. For these training instances, $\tau_{Hi} = 0.609$ resulted in the highest accuracy.

The value of τ_{Lo} is given by

$$\tau_{Lo} = \epsilon \cdot \tau_{Hi} \tag{35}$$

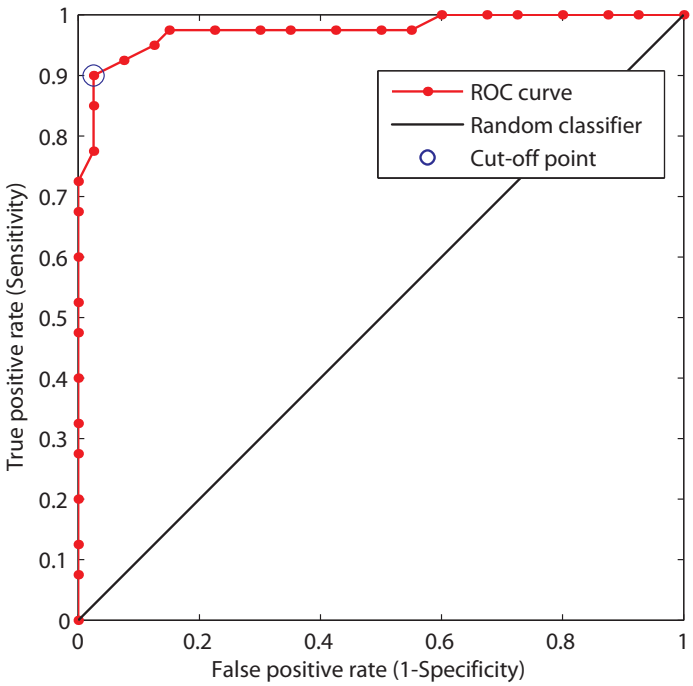


Figure 74: ROC curve for the 160 training instances.

Table 3: Confusion matrix for the 160 training instances. $\tau = 0.609$, Accuracy (ACC)= 0.937, Area under the ROC curve (AUC)= 0.974, True Positive Rate (TPR)= 0.9, Specificity (SPC)= 0.975.

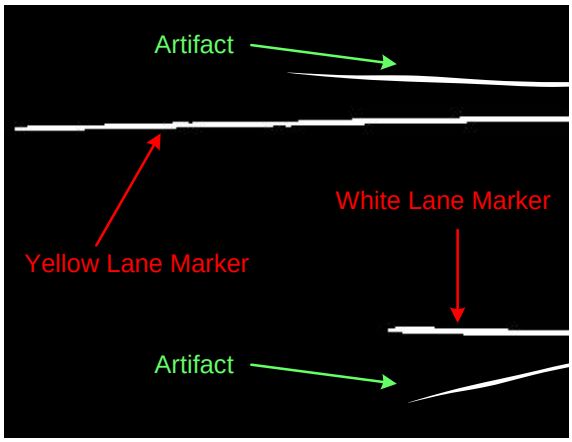
		Predicted Outcome		
		Positive	Negative	Total
Input	Positive	72	8	80
	Negative	2	78	80
	Total	74	86	160

where $\epsilon \in (0, 1)$ is a multiplication factor. The value of ϵ could not be determined analytically; hence, we tested ϵ with several empirically determined values and observed that $\epsilon = 0.95$ produced the best result. The values used in testing are listed in Sec. 2.5.7 and Appendix B. The Hysteresis threshold is denoted by the notation $\geq_H \ (\tau_{Hi}, \tau_{Lo})$, where τ_{Hi} and τ_{Lo} are the high and low thresholds, respectively. For example in

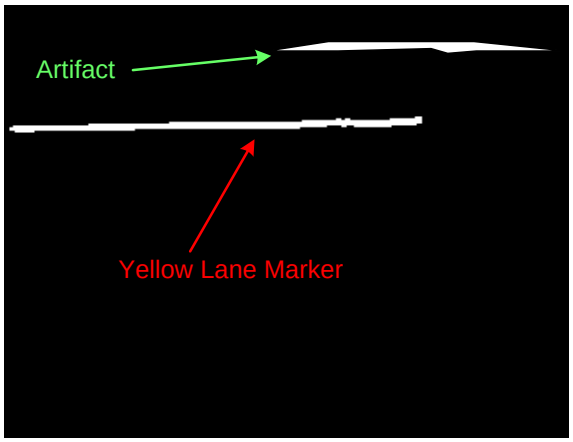
$$Map_{Gr} \geq_H (\tau_{Hi}, \tau_{Lo}) = B_{Res} \tag{36}$$

Map_{Gr} is a grayscale image and B_{Res} is the resultant binary image generated by application of the Hysteresis threshold [35].

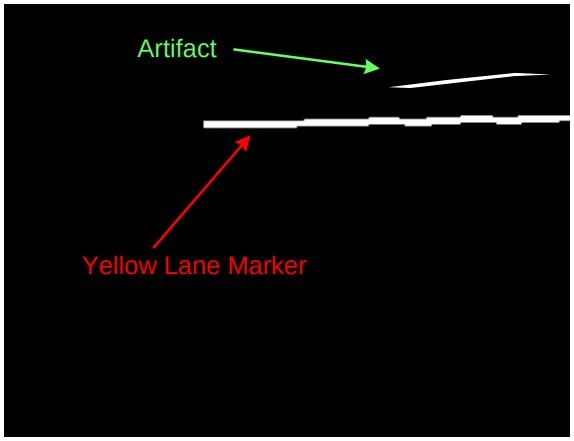
Since $\hat{W}_n$ consists of three color channels, and we have defined three templates in Fig. 69, Template matching and the Hysteresis threshold are performed on each color channel using each of the three templates, which results in nine binary images at the end of the Filtering block. In Fig. 75, we have shown three binary images, which are created as a result of performing Template matching followed by the Hysteresis threshold between the normal lane marker template and each color channel of $\hat{W}_n$. Referring back to Sec. 2.5.2, we had observed that the white lane markers are prominently visible in the Y channel of $\hat{W}_n$ and yellow lane markers are prominently



(a) Y channel.



(b) Cb channel.

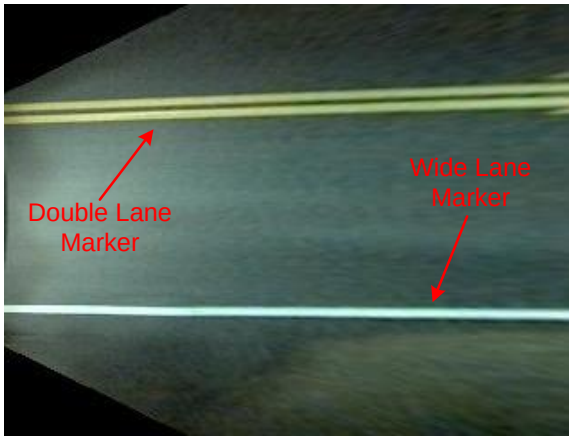


(c) Cr channel.

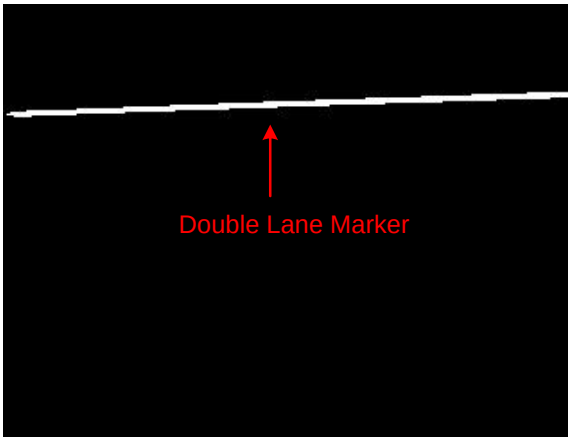
Figure 75: Template matching with a normal lane marker template followed by the application of the Hysteresis threshold on each channel.

visible in the Cb and Cr channels; as a result, in Fig. 75 the white lane markers are

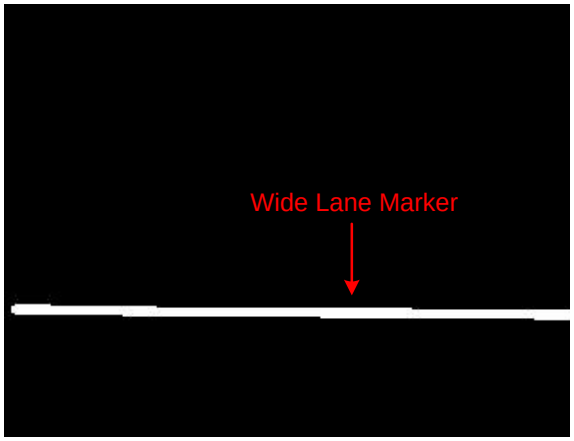
detected in binary image of the Y channel and the yellow lane markers are detected in binary images of the Cb and Cr channels [35]. However, the binary images also contain artifacts as shown in Fig. 75. The coefficients maps created using the wide and double lane marker templates contained mostly low valued coefficients. This resulted in their corresponding binary images to be mostly black since the wide or the double lane markers are not present in $\hat{W}_n$; hence, these binary images are not shown in Fig. 75. However, in Fig. 76, we have shown a world image that contains wide and double lane markers and their resultant binary images created as a result of Template matching followed by Hysteresis threshold.



(a) World image containing double and wide lane markers.



(b) Double lane marker detected.



(c) Wide lane marker detected.

Figure 76: Template matching with double and wide lane marker templates followed by Hysteresis threshold.

2.5.4 Lane Region Merging

The previous filtering block generates binary images in which the lane markers have been detected. However, these images may also contain false detections. Therefore, performing a simple logical AND or OR across all nine binary images does not provide accurate results. The goal of this block is to combine the nine binary images in a way that preserves the lane markers while minimizing the artifacts [35]. Prior to doing this, some notation needs to be defined. Let $B_{Cb,N}$ be the binary image that is the result of matching a normal lane template with the Cb channel of the $\hat{W}_n$ followed by hysteresis thresholding. Similarly, let $B_{Cr,N}$ and $B_{Y,N}$ be the binary images created as a result of matching the same template with Cr and Y channels, respectively. $B_{Cb,N}$ and $B_{Cr,N}$ are shown in Fig. 75b and 75c, respectively.

Yellow lane marker regions are often visible in the binary images of both Cb and Cr channels. As a result, these two binary images need to be combined to produce a single binary image. Using the notation described above, a binary image ($B_{Yel,N}$) containing only normal and yellow lane markers is created as follows. First

$$(B_{Cb,N} \wedge B_{Cr,N}) = B_{\wedge} \tag{37}$$

$$(B_{Cb,N} \vee B_{Cr,N}) = B_{\vee} \tag{38}$$

where $B_{\wedge}$ is a logical AND operation that is performed on a pixel-by-pixel basis between corresponding pixels in $B_{Cb,N}$ and $B_{Cr,N}$. Similarly, $B_{\vee}$ is a logical OR operation between corresponding pixels in $B_{Cb,N}$ and $B_{Cr,N}$. The truth tables for a logical AND operation and a logical OR operation are provided in Tables 4 and 5, respectively. Next

$$B_{\wedge} + B_{\vee} \rightarrow Img_{Gr} \geq_H (2,1) = B_{Yel,N} \tag{39}$$

where Img_{Gr} is a grayscale image as shown in Fig. 77 that is created by pixel-wise adding $B_{\wedge}$ and $B_{\vee}$. In Img_{Gr}, the addition leads to overlapping lane marker pixels

Table 4: Truth table for logical AND between two inputs p and q.

Input		Output
p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Table 5: Truth table for logical OR between two inputs p and q.

Input		Output
p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

from $B_{\wedge}$ and $B_{\vee}$ summing to a value of 2 (red pixels). The non-overlapping lane pixels produce a value of 1 (green pixels), and background pixels are 0. The application of

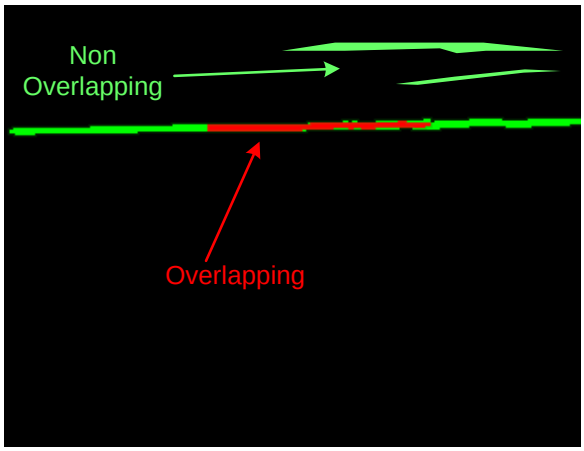


Figure 77: Overlapping pixels in Img_{Gr} shown in red and non-overlapping pixels shown in green.

the Hysteresis threshold in Eq. (39) helps join the "connected" high and low threshold regions from $B_{Cb,N}$ and $B_{Cr,N}$ [35] as shown on in Fig. 78. These connected regions often correspond to yellow lane markers. If only a logical AND was used to combine the two binary images, the resultant image would contain only the overlapping regions (red pixels) as shown in Fig. 79a, while a logical OR would result in some stray low

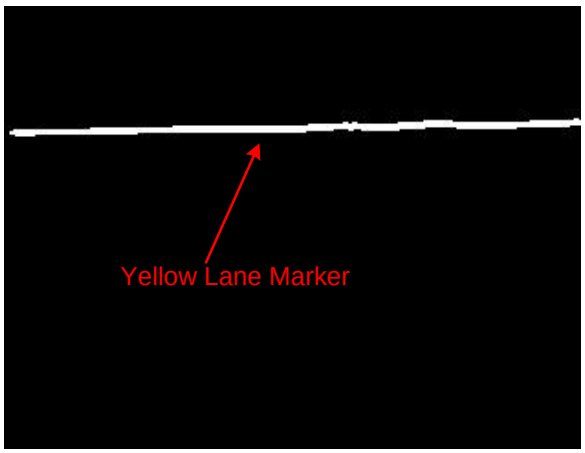
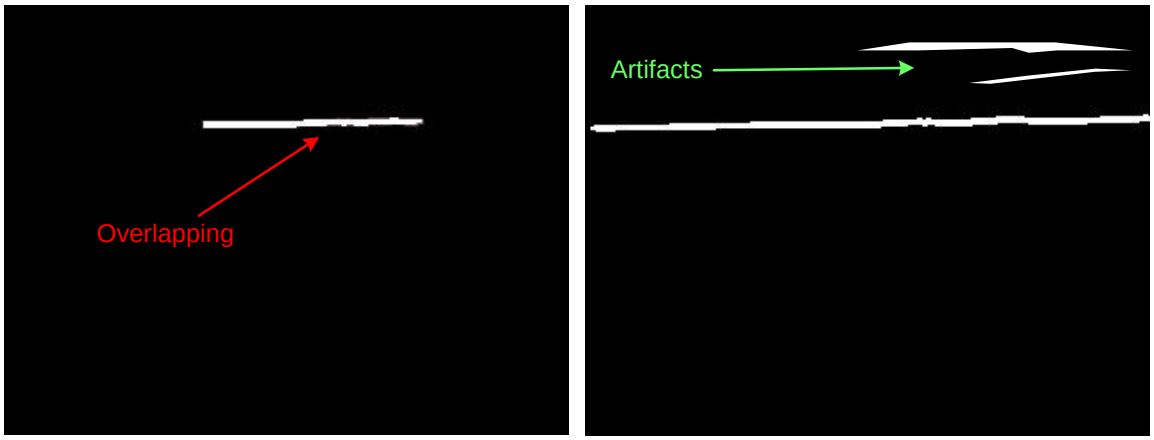


Figure 78: $B_{Yel,N}$ is created by applying threshold threshold to Img_{Gr}.

threshold regions to also be included as shown in Fig. 79b.



(a) $B_{\wedge}$, logical AND between $B_{Cb,N}$ and $B_{Cr,N}$. (b) $B_{\vee}$, logical OR between $B_{Cb,N}$ and $B_{Cr,N}$.

Figure 79: The result of logical operations between $B_{Cb,N}$ and $B_{Cr,N}$.

A binary image containing white lane markers of normal thickness is created as follows:

$$B_{Y,N} \wedge \neg(B_{Yel,N} \oplus SE) = B_{Whi,N} \tag{40}$$

where $\oplus$ is a morphological dilation, and $\neg$ is a logical NOT. In Eq. (40), $B_{Y,N}$ is a binary image containing only lane markers in the Y channel and $B_{Yel,N}$ is the image containing the detected yellow lane markers from Eq. (39) and shown in Fig. 78. First, $B_{Yel,N}$ is dilated with a structuring element SE whose dimensions are $k \times 1$ pixels. Here, $k = 80$ pixels which is the equivalent of 10 ft in $\hat{W}_n$. Next, $B_{Yel,N}$ is

dilated with SE. The dilated image ($B_{Yel,N} \oplus SE$) is negated by a logical NOT to create a mask as shown in Fig. 80.

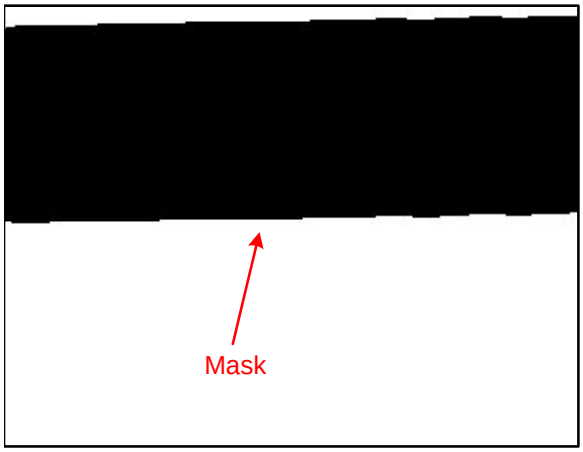


Figure 80: Mask used to mask out detected pixels in $B_{Yel,N}$.

The truth table for the logical NOT operation is provided below

Table 6: Truth table for logical NOT for input p.

Input	Output
p	$\neg p$
0	1
1	0

Then, the $\wedge$ operation between $B_{Y,N}$ and $\neg(B_{Yel,N} \oplus SE)$ produces the binary image $B_{Whi,N}$ that contains only white lane markers as shown in Fig. 81. The purpose of $\neg(B_{Yel,N} \oplus SE)$ is to mask out areas from $B_{Yel,N}$ that have been detected as yellow lane markers. By dilating $B_{Yel,N}$ with a 10 foot wide structuring element, nearby areas that are within 5 ft on either side of yellow lane marker pixels are masked out. This type of masking prevents the re-detection of the yellow lane marker pixels in $B_{Y,N}$. Lastly, the two thresholded images obtained from Eq. (39) and (40) are combined to produce a binary image containing only normal lane markers

$$B_{Whi,N} \vee B_{Yel,N} = B_N \tag{41}$$

which is denoted by B_N and shown in Fig. 82. Eq. (37)-(41) are similarly used to

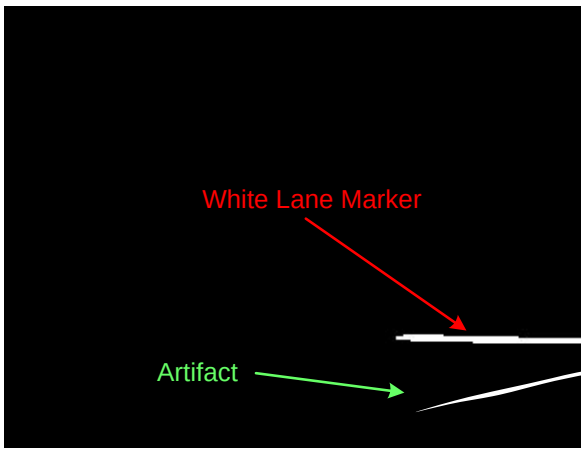


Figure 81: $B_{Whi,N}$, binary image containing only white lane markers.

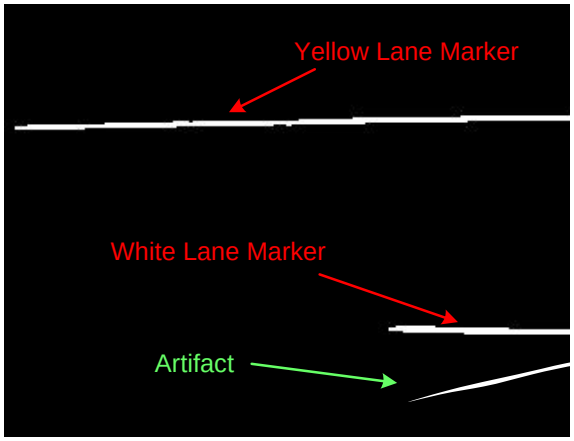


Figure 82: B_N, binary image with narrow lane markers and few artifacts.

create binary images containing double (B_D) and wide (B_W) lane markers. Finally, the three binary images are merged

$$B_W \vee B_N \vee B_D = B_{Lane} \tag{42}$$

to create B_{Lane}, a binary image containing only lane markers (both white and yellow markers) with few artifacts [35]. Since no wide or double lane markers are present in the image, B_{Lane} is equal to B_N in this example.

2.5.5 Lane Marker Classifier

The output of the Lane Region Merging block is a binary image B_{Lane} in which clusters of pixels or objects corresponding to lane markers are visible. However, if multiple

objects are visible in B_{Lane}, it is not trivial to decide which objects should be kept and which should be ignored. An example is shown in Fig. 82 where multiple objects are present in the image. However, some of these objects correspond to artifacts. Therefore, the goal of the Lane Marker Classifier is to keep objects corresponding to lane markers while eliminating objects corresponding to artifacts [40]. As shown in Fig. 83, the Lane Marker Classifier consists of two stages:

1. Ellipse modeling.

2. Windowing.

To perform ellipse modeling, each discrete object in Fig. 82 needs to be classified as either a solid or dashed lane marker. The FHA classifies a lane marker as either

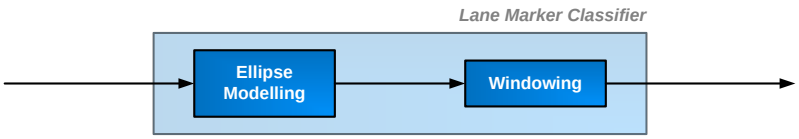


Figure 83: Stages in the Lane Marker Classifier block.

solid or dashed based on its length [19]. In world images, lane markers appear to move from the right to the left side of the image; additionally, they may be slightly curved as shown by the yellow lane marker in Fig. 82. Therefore, an object in B_{Lane} is classified based on its horizontal length. If the horizontal length exceeds 10 ft, the object is classified as a solid lane marker. Otherwise, the object is classified as a dashed lane marker. For illustration, solid lane markers are shown in red and dashed lane markers are shown in cyan in Fig. 84. A dashed lane marker is fit inside an ellipse and the major axis is determined. The major axis of the ellipse provides an estimate of the orientation of the lane marker as shown in Fig. 84. As a solid lane marker may be curved, fitting an ellipse to the entire length of the object may not always provide an estimate of the orientation of the lane marker. For a solid lane marker, the procedure to estimate the major axis is modified slightly. Hence, only

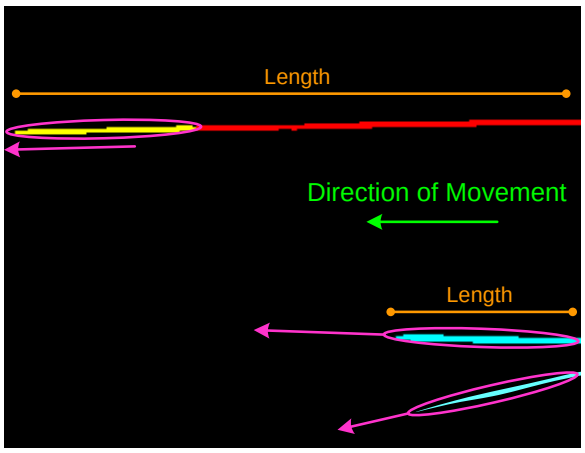


Figure 84: Ellipse fitted to each object in B_{Lane}.

the leading 10ft (shown in yellow) of the solid lane marker is fit inside the ellipse and is used in direction estimation as shown in Fig. 84.

Initially, this direction estimation was performed by fitting a line through the object using the Hough transform [22]. Since the objects in B_{Lane} are generally rectangular in shape, the path along the diagonal of the object is selected as it contains the maximum number of pixels. However, as shown in Fig. 85, this choice is often incorrect [40].

(a) Best fitting lines using Hough transform.

(b) Direction of the major axis of the ellipse.

Figure 85: Determining the orientation of the lane marker.

Next, in windowing, the major axis of each ellipse must meet certain requirements of a window to be classified as a lane marker [40]. To create the window, we take the ground truth of the left lane boundary for images in which the vehicle is travelling in the middle of straight and curving roads. Then, we transform this ground truth to world co-ordinates as explained in Sec. 2.4.7. A number of transformed ground truths

appear as red curves in Fig. 86. Next, we empirically determine a double parabolic

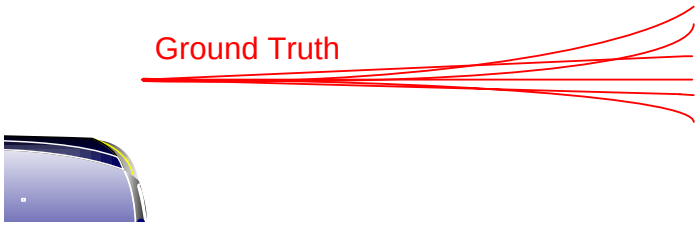


Figure 86: Ground truth transferred to world co-ordinates.

window inside which each of the ground truth curves are present. This parabolic window is shown in Fig. 87. However, it is very unlikely that the vehicle will always be traveling in the middle of the road at all times. Hence, to accommodate cases

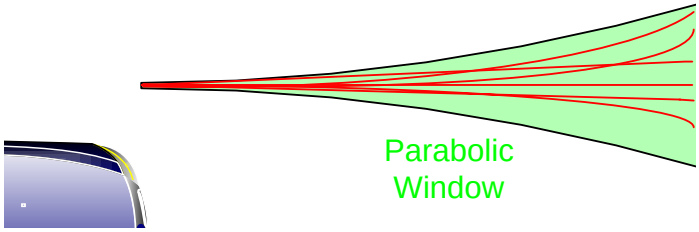


Figure 87: Empirically designed parabolic window.

when the vehicle is traveling near or away from one or more lane boundaries, this window is widened. Widening is performed simply by moving the two boundaries up and down as shown in Fig. 88. Additionally, widening allows the ground truth to be present within the window even when the vehicle may not be traveling in the middle

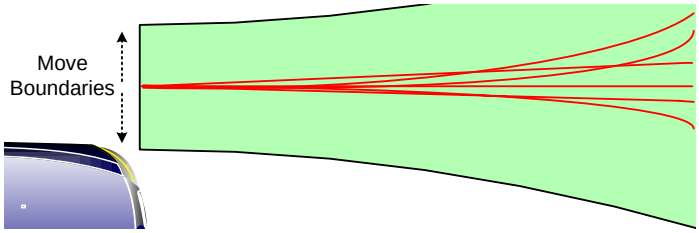


Figure 88: Parabolic window is widened by moving the boundaries.

of the road.

The equation for each boundary of the window is

$$y = y_L \pm (0.0078x^2 + b) \qquad \text{where } x \in [0, \delta] \tag{43}$$

where $y_L = 6$ ft is the location of the left lane boundary with respect to the vehicle assuming that the vehicle is traveling in the middle of a road where the lane boundaries are 12 ft apart as shown in Fig. 89. Additionally, $b = 4$ ft is the offset used to widen

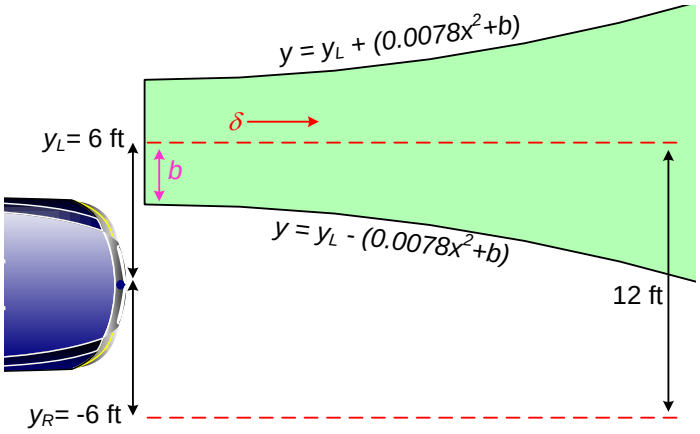


Figure 89: Specifications of the parabolic window.

the window and δ is the length of the window. The value 0.0078 in Eq. (43) is used to compress the parabolas. It is the smallest value we could use such that the ground truth curves from Fig. 87 are present inside the parabolic window that has a length of 50 ft. A similar window is created on the right side of the vehicle and the equation for each of its boundaries is defined as

$$y = y_R \pm (0.0078x^2 + b) \qquad \text{where } x \in [0, \delta] \tag{44}$$

where $y_R = -6$ ft is the location of the right lane boundary with respect to the vehicle. As a result, we have two windows, one on either side of the vehicle within which the ground truth will be present at most times. Furthermore, the ground truth will be present when the vehicle travels near or away from a lane boundary, on both straight and curving roads where the lanes are $8 - 20$ ft wide. We complete the creation of the window by assigning a length δ to it. However, we were unable to determine the best length for the window analytically; therefore, we tested δ with several empirically determined values and selected the one that produced the best result. The values

used in testing are listed in Sec. 2.5.7 and Appendix B. Finally, the different values for δ result in different shapes of the windows as shown in Fig. 90.

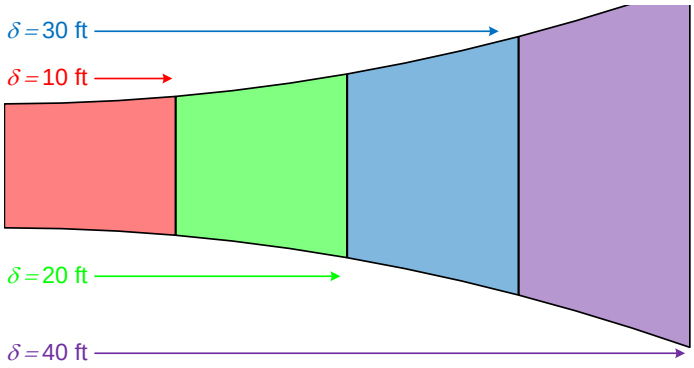


Figure 90: Different values of δ result in different window shapes.

As mentioned before, each object in B_{Lane} is fitted inside an ellipse. Therefore, for an object to be classified as a lane marker, it must meet two requirements:

1. The extension of the ellipse's major axis must pass through both ends of the window as shown in Fig. 91. The two ends of each window are highlighted in cyan in Fig. 91. We specify this requirement since we observed that both the solid and the dashed lane markers meet this requirement in the ground truth.

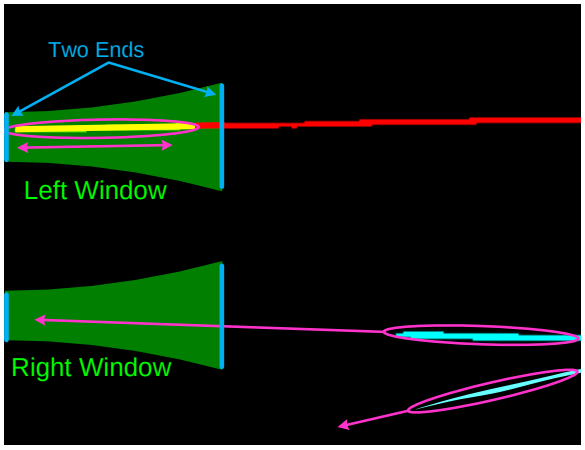
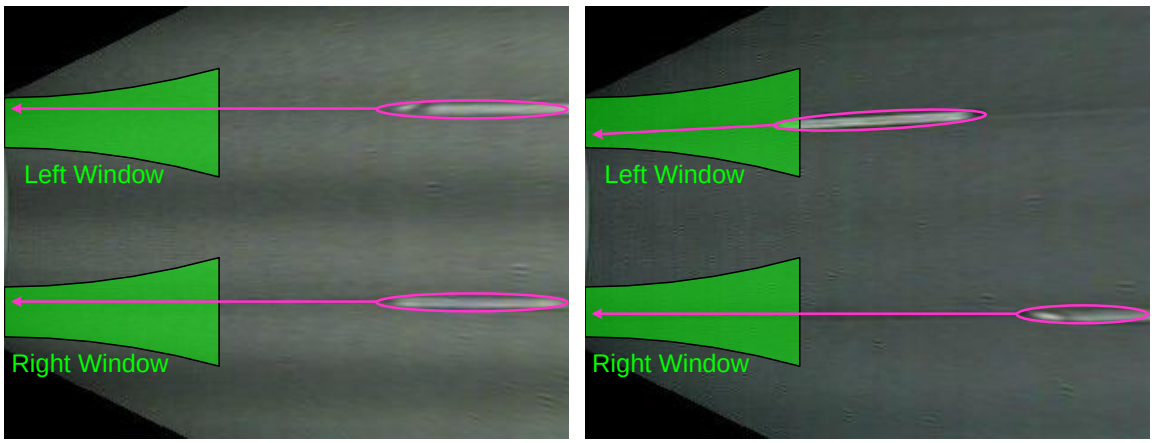


Figure 91: Major axes pass through both ends of the window for lane objects.

The images in Fig. 92 support this claim. Therefore, lane markers in B_{Lane} should also meet this requirement. On the other hand, most artifacts do not meet this requirement, and hence, are eliminated.



(a) On a straight road. (b) On a curving road.

Figure 92: Major axes pass through the windows on different road conditions.

2. The major axis of its ellipse has an angle $\theta \leq |\alpha|$ with respect to the horizontal axis. θ is measured as shown in Fig. 93. α is determined by approximating the parabolic window as a regular trapezoid and determining the base angle of the shorter base with respect to the horizontal as shown in Fig. 94. During testing we tried different values for δ that resulted in different window shapes; therefore, α needs to be recalculated every time the window length δ changes. This angle restriction helps to eliminate objects that have a predominantly vertical orientation e.g. the artifact near the bottom right of Fig. 82.

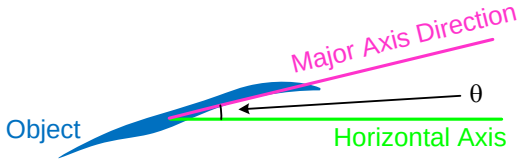


Figure 93: Angle of the major axis.

Finally, an object is classified as a left lane marker if the major axis of its ellipse passes through both ends of the left window [40]. Similarly, an object is classified as a right lane marker if the major axis of its ellipse passes through both ends of the right window.

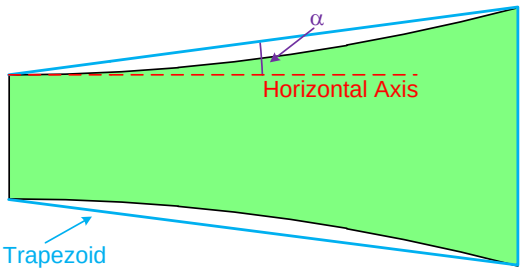


Figure 94: Approximating a trapezoid to determine α.

We also tested with semi-circular, triangular, and rectangular window shapes. A rectangular window also yields promising results. However, selecting the appropriate width for the window is problematic. This is illustrated with the image in Fig. 95 where two widths for rectangles are compared. The inner rectangle has a width of

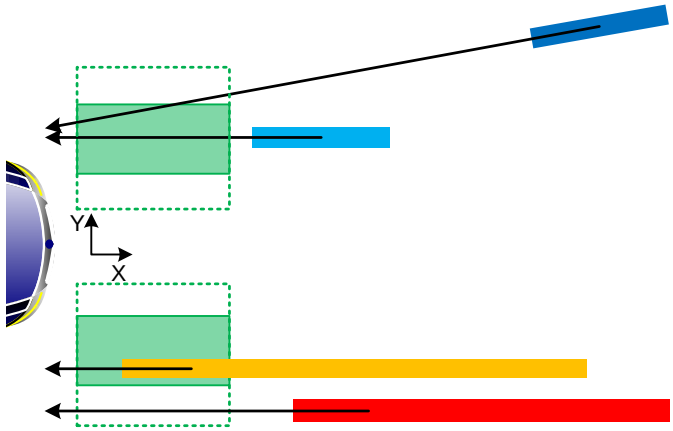


Figure 95: Comparing two widths for rectangular windows. The inner rectangle has a width of 7 ft and the outer dotted rectangle has a width of 14 ft. The light blue object is a dashed lane marker on a straight road. The dark blue object is a dashed lane marker on curving road. The orange object is a solid lane marker. The red object represents a false detection such as a sidewalk.

7 ft and the outer dotted rectangle has a width of 14 ft. Furthermore, the light blue object represents a dashed lane marker on a straight road while the dark blue objects represents a dashed lane marker on curving road. Similarly, the orange object represents a solid lane marker while the red object represents a false detection such as a sidewalk that is initially classified as a solid lane marker. Although the inner

rectangle eliminates the red object, it also ignores the dark blue object since the major axis of its ellipse does not pass through both ends of the window. On the other hand, the outer rectangle retains the dark blue object; but it also retains the red object corresponding to the artifact. In contrast, the parabolic shape provides a blend between both sizes of rectangular windows.

Using the Lane Marker Classifier, objects in B_{Lane} corresponding to lane markers are correctly identified [40]. This is shown in Fig. 96 where the artifact in Fig. 82 is removed at the end of the Lane Marker Classifier block. Furthermore, objects

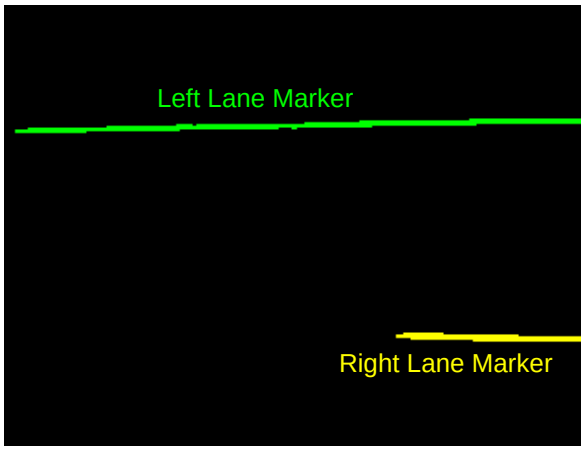
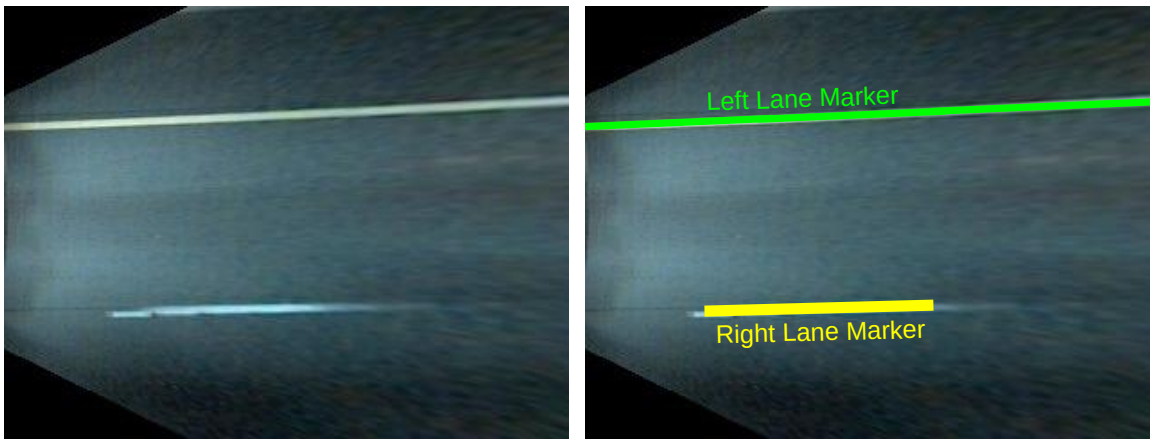


Figure 96: Objects in B_{Lane} classified as left and right lane markers.

classified as left lane markers are shown in blue and objects classified as right lane markers are shown in yellow in Fig. 96. In Fig. 97 and 98, we have shown images where the Lane Marker Classifier has correctly identified the left and right lane markers.

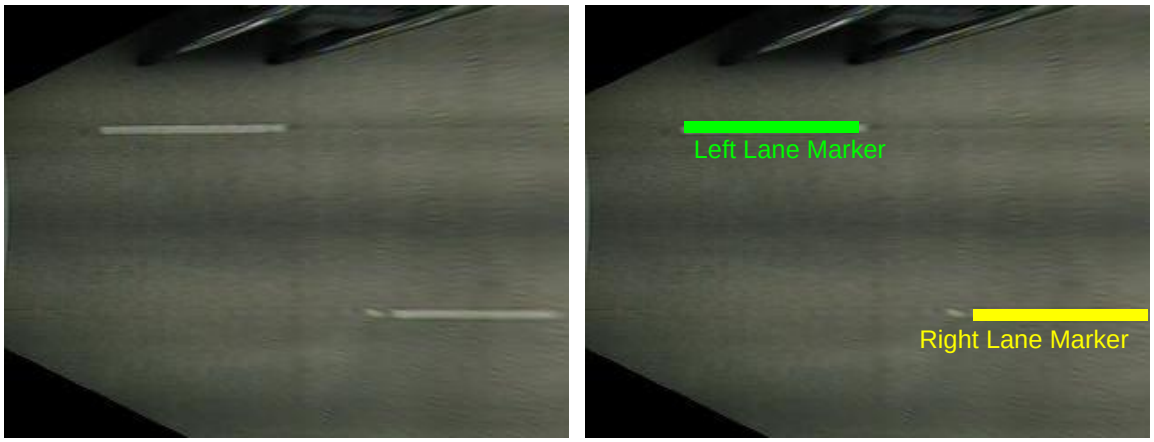
2.5.6 Piecewise Tracking

The Piecewise Tracker described here is used to estimate the lane boundary movement from one image to the next. However, unlike a common tracker that is described in Sec. 2.4.7 that uses a single Kalman filter to estimate the parameters of the curve on which lane markers may lie, a Piecewise Tracker uses several Kalman filters to estimate the location of the curve in various parts of the image. The Piecewise Tracker is useful in estimating the shape of the lane boundary when it is difficult to



(a) Original image. (b) Detected lane markers.

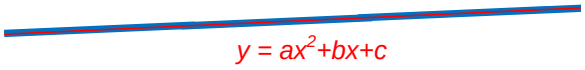
Figure 97: The Lane Marker Classifier is able to detected lane markers on a curving road.



(a) Original image. (b) Detected lane markers.

Figure 98: The Lane Marker Classifier is able to detected lane markers on a straight road.

describe the shape analytically. For example, a solid lane marker is long and covers most of width of the world image from left to right. Hence, after it is detected in B_{Lane}, a curve representing the solid lane marker can be described easily as shown in Fig. 99a. On the other hand, dashed lane markers are small segments and they can lie on one of many curves as shown in Fig. 99b. Here, the Piecewise Tracker observes the lane marker movement over a sequence of images and estimates the shape of the lane boundary or curve on which it lies.



(a) Lane marker shown in purple and the curve shown in red. The solid lane marker lies on a single curve.

(b) Lane marker shown in purple and the curve shown in red. Dashed lane marker can lie on one of many curves shown by the analytic equations.

Figure 99: Estimating the curve on which lane markers lie.

To explain the Piecewise Tracker, let us take a look at B_{Lane} from Fig. 96; however, we only consider the left lane markers for illustration. Next, we specify Sampling Columns as we did in Sec. 2.4.4 and shown in Fig. 100. As before, the columns are spaced 24 pixels apart in B_{Lane}. Additionally, each column is labeled as C_i, where i is the index of the column. Next, we observe the intensity profile inside

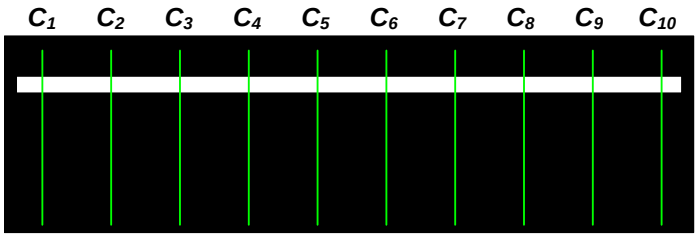


Figure 100: Sampling Columns in the binary image.

each column. The portion of the column corresponding to a lane marker appears as a pulse of amplitude 1 as shown in Fig. 101. The lane marker appears as a pulse rather than a spike because the lane markers have a thickness of a few pixels in B_{Lane}. Therefore, the mean location of the binary "1" pixels is computed and used as the best representation of the lane marker location in that column as shown in Fig. 101. If multiple pulses exist in the column, then we simply compute the mean of the

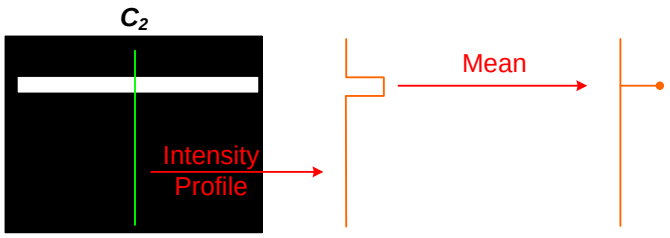


Figure 101: Intensity profile of a lane marker and the mean location the binary "1" pixels.

widest pulse. This approach helps avoid interference from small stray objects that may be running alongside the lane marker. The computed mean location serves as the measurement that is used by the Kalman filter. However, this measurement is used to estimate the position of the lane boundary only in a particular Sampling Column in the images of the video clip. Since we have multiple Sampling Columns, multiple Kalman filters are used to estimate the position of the lane boundary in the respective columns. Therefore, the Sampling Columns serve as the parts of the image where the lane boundary is estimated.

The state vector for the Kalman filter in each column C_i is

$$\mathbf{x}_{n,C_i} = \begin{bmatrix} y_{n,C_i} & \dot{y}_{n,C_i} \end{bmatrix}^T \tag{45}$$

where y_{n,C_i} is the computed mean location of the pulse in C_i, and $\dot{y}_{n,C_i}$ is its derivative. For a constant velocity model, the state equation can be written as

$$\mathbf{x}_{n+1,C_i} = \mathbf{A}\mathbf{x}_{n,C_i} + \mathbf{w}_n \tag{46}$$

$$\mathbf{x}_{n+1,C_i} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \mathbf{x}_{n,C_i} + \mathbf{w}_n \tag{47}$$

where $\Delta T = 33$ms and the process noise vector $\mathbf{w}_n \sim N(0, \mathbf{Q})$ with

$$\mathbf{Q} = \begin{bmatrix} \sigma_y^2 & \sigma_y^2 \cdot \frac{2}{\Delta T} \\ \sigma_y^2 \cdot \frac{2}{\Delta T} & \sigma_y^2 \cdot \frac{4}{\Delta T^2} \end{bmatrix} \tag{48}$$

where σ_y^2 is the variance in process noise associated with y. The variance σ_y^2 was determined experimentally as follows. We took three sequential images from a video

clip that were indexed as n, $n+1$, and $n+2$. Each image contains ground truth (see Chap. 4 for details on the ground truth). Then, for each image, the ground truth curve was transformed to world co-ordinates (transformation is described in Sec. 2.4.8) and its location is observed in a particular Sampling Column, say C_5. Thus, we have three locations y_{n,C_5}, y_{n+1,C_5}, and y_{n+2,C_5} of the curve for the three images that refer to the ground truth. Then, $\dot{y}_{n+1,C_5}$, the derivative of y_{n,C_5} with respect to time in image $n+1$, is computed as

$$\dot{y}_{n+1,C_5} = \frac{y_{n+1,C_5} - y_{n,C_5}}{\Delta T} \tag{49}$$

then,

$$\hat{y}_{n+2,C_5} = y_{n+1,C_5} + \dot{y}_{n+1,C_5} \cdot \Delta T \tag{50}$$

and the error is computed as

$$\text{error} = y_{n+2,C_5} - \hat{y}_{n+2,C_5} \tag{51}$$

This error computation is performed on 100 different image triplets and used to create a distribution. Finally, assuming that the error has a zero mean Gaussian distribution [33], the standard deviation of the distribution is computed to give $\sigma_y = 0.11$.

The measurement equation is

$$z_{n,C_i} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_{n,C_i} + v_n \tag{52}$$

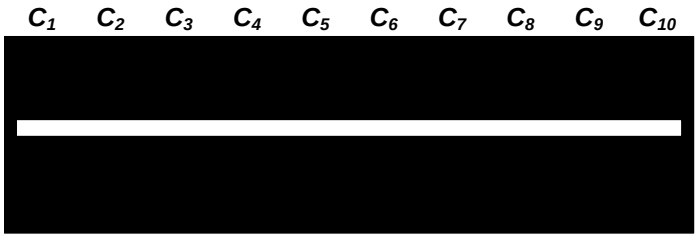
where the measurement noise vector $v_n \sim N(0, \sigma_z^2)$. The variance σ_z^2 was determined experimentally as follows. We took 100 images that contained ground truth and observed the location of the lane boundary curve in C_5 as mentioned above. Thus, we have 100 locations of the curve y_{1,C_5}, y_{2,C_5} ... for the 100 images. Then, we performed lane detection using the ALD 2.0 up to and including the Lane Marker Classifier block from Fig. 64 on each of the 100 images and observed the computed mean location of the pulse $\hat{y}$ (because y is the ground truth and $\hat{y}$ is the measurement);

therefore, we have $\hat{y}_{1,C_5}$, $\hat{y}_{2,C_5}$ ... for the 100 images. The error is computed as

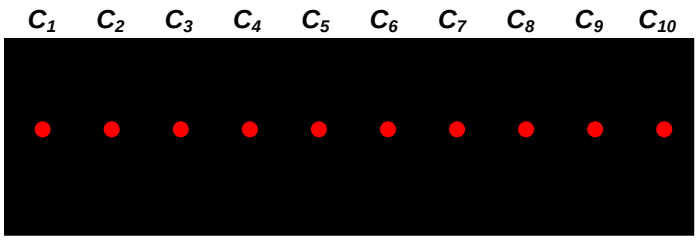
$$\text{error} = y_{1,C_5} - \hat{y}_{1,C_5} \tag{53}$$

and a distribution is created from the collection of error values. Again, assuming that the error has a zero mean Gaussian distribution [33], the standard deviation of the distribution is computed. This computation gives $\sigma_z = 0.121$. This concludes setting up the parameters and the matrices for the Kalman filters. Both $\mathbf{Q}$ and σ_z^2 are assumed to be the same in each Sampling Column. Finally, each error covariance matrix is initialized with a large value on its diagonal.

Let us look at a simple example to illustrate how the Piecewise Tracker works. Consider a solid lane marker in B_{Lane} as shown in Fig. 102a. This marker extends from the left to right side of the image. Hence, a measurement is available in every Sampling Column. Thus, these measurements can be used to estimate the lane boundary position in the respective columns as shown in Fig. 102b. For a dashed



(a) Solid lane marker.

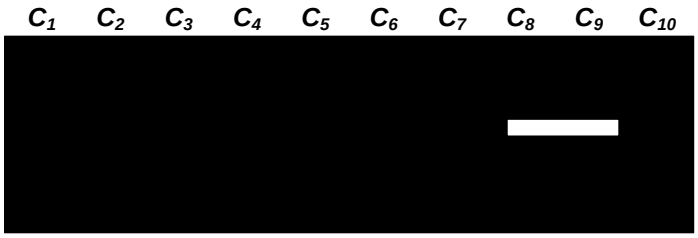


(b) Estimates in the Sampling Columns.

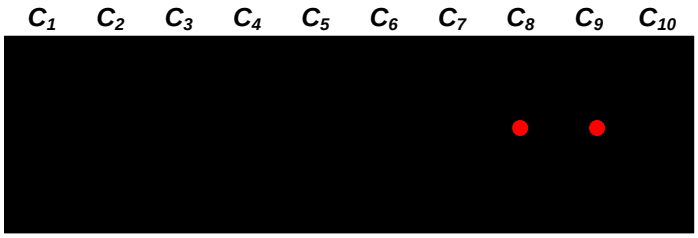
Figure 102: Estimating the lane boundary.

lane marker, the approach is slightly different. Consider $B_{Lane,n}$ (notice the appended subscript "n") in Fig. 103a that corresponds to image n in the video clip. The dashed

lane marker only resides in a few columns, in this case C_8 and C_9. Therefore, measurements from C_8 and C_9 can be used to estimate the position of the lane boundary in C_8 and C_9. The estimated positions of the lane boundary in C_8 and C_9 are shown in Fig. 103b. In the next image, $B_{Lane,n+1}$, the lane marker has moved to the left and



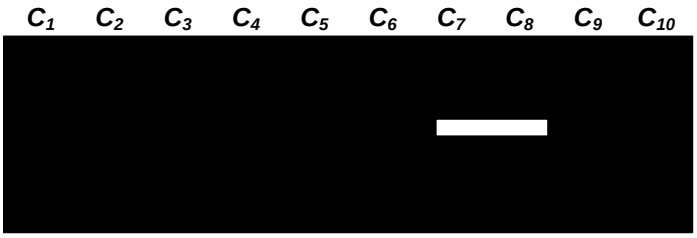
(a) Dashed lane marker in C_8 and C_9 of $B_{Lane,n}$.



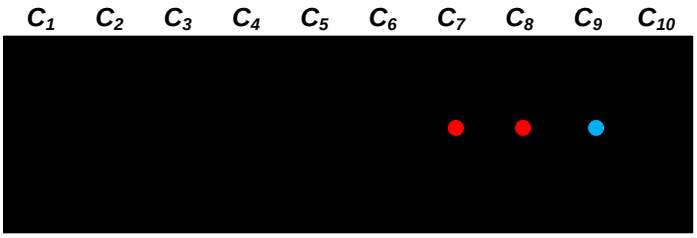
(b) Estimates in a few Sampling Columns.

Figure 103: Estimating the lane boundary.

now resides in C_7 and C_8 as shown in Fig. 104a. Hence, measurements from C_7 and C_8 are used to produce the estimate of the lane boundary in C_7 and C_8. However, no measurement is available in C_9. Therefore, the position of the lane boundary in C_9 is estimated only through prediction as shown in Eq. 28. The estimated positions of the lane boundary in $C_7 - C_9$ are shown in Fig. 104b. In the next image, $B_{Lane,n+2}$, the lane marker has moved to the left yet again and now resides in C_6 and C_7 as shown in Fig. 105a. Both C_6 and C_7 use measurements from the columns to produce their estimates. However, no measurements are received from C_8 and C_9. Therefore, the position of the lane boundary in C_8 and C_9 is estimated again using only predictions. The estimated positions of the lane boundary in $C_6 - C_9$ are shown in Fig. 105b. After a few images, when the lane marker has moved to the left and out of the image, estimates for the lane boundary will exist in all Sampling Columns C_i and can be

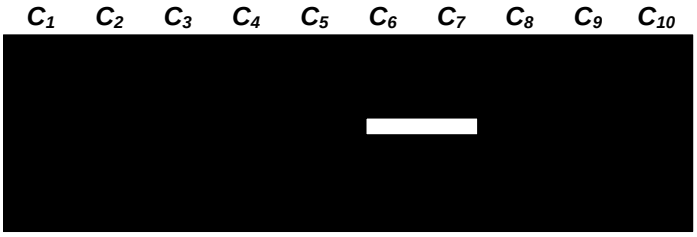


(a) Dashed lane marker in C_7 and C_8 of $B_{Lane,n+1}$.

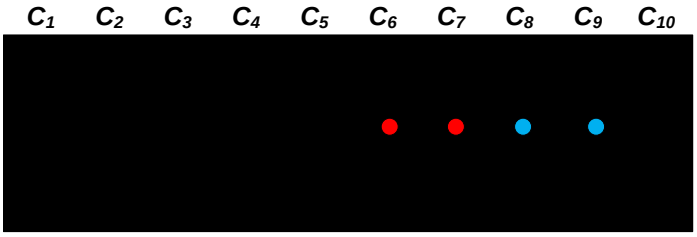


(b) Estimates made using measurements are shown in red. Estimates made using only predictions are shown in blue.

Figure 104: Estimating the lane boundary.



(a) Dashed lane marker in C_6 and C_7 of $B_{Lane,n+2}$.



(b) Estimates made using measurements are shown in red. Estimates made using only predictions are shown in blue.

Figure 105: Estimating the lane boundary.

used to approximate a curve representing the lane boundary as shown in Fig. 106.

As in Sec. 2.4.7, an estimate using only prediction is allowed to be produced a maximum of 60 times. After 60 sequential predictions, the Kalman filter is disabled and no estimate is produced in that Sampling Column. This limit on the sequential

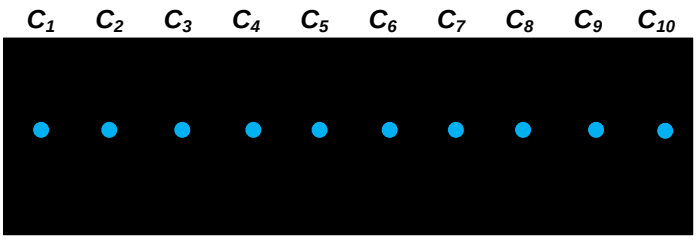


Figure 106: Estimates in each Sampling Column are made using only prediction.

predictions prevents an incorrect estimate from being produced in case no lane marker is present on the road. Finally, if the Kalman filter in a column has been disabled and a measurement is available then the Kalman filter is reinitialized.

Once we have estimates in at least three Sampling Columns, RANSAC (details in Sec. 2.4.6) [23] is used to eliminate outliers among the estimates. Then, LLS (details in Sec. 2.4.6) [32] is used to fit a curve to the inliers to represent the lane boundary as shown in Fig. 107. In the ALD 2.0, LLS uses a quadratic model instead

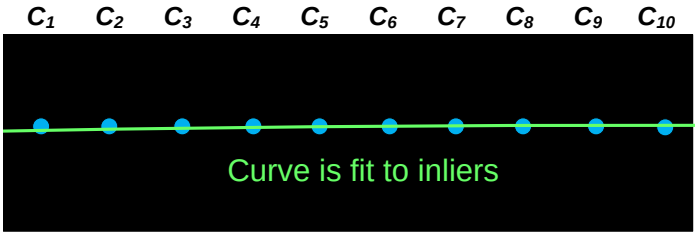


Figure 107: Fitting a curve to the inliers using RANSAC followed by LLS.

of the straight line model used in the ALD 1.0. As described in Sec. 2.4.6 LLS is sensitive minor perturbations in the inliers. However, we observed that the Piecewise Tracker often gets rid of most perturbations within the estimates that is produces. This elimination of pertubations allows the LLS to accurately produce a curve that represents the lane boundary. Finally, the curve undergoes a perspective mapping to show its representation of the lane boundary in camera image $I(n)$. Details of perspective mapping can be found towards the end of Sec. 2.4.7. The camera image with the estimated lane boundary is shown in Fig. 108.

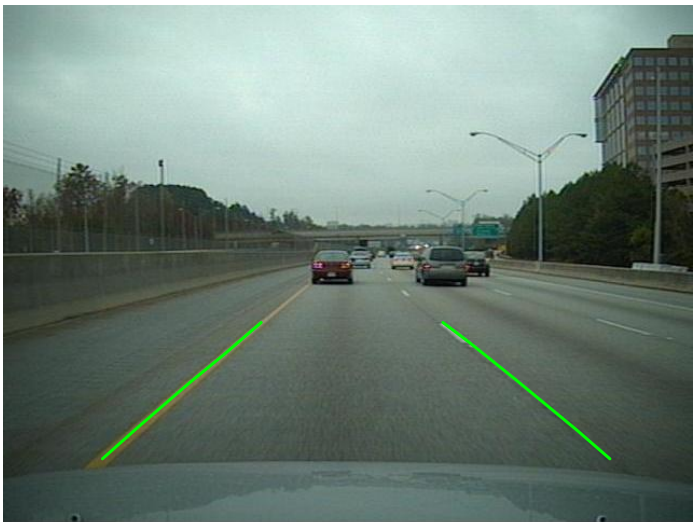


Figure 108: Estimated lane boundaries after tracking.

Since tracking is performed separately on left and right lane boundaries, a Piecewise Tracker is implemented and used to estimate the position of the right lane boundary. Although we create separate state vectors for the estimates in each Sampling Column, it is possible to encapsulate all the estimates into a single state vector. However, estimates corresponding to the disabled Kalman filters need to be masked out from the state equation. This results in a state vector whose length changes dynamically depending on the number of Kalman filters that are actively producing estimates in the Sampling Columns. Hence, a single state vector is undesirable [40].

2.5.7 Performance Evaluation

To evaluate the ALD 2.0, we use the same performance measures that were described in Sec. 2.4.9. In addition, we also use the same eight video clips that were used to test the ALD 1.0. Two parameters of the ALD 2.0 ϵ, the multiplication factor of the high threshold τ_{Hi} that was described in Sec. 2.5.3, and δ, the length of the window that was described in Sec. 2.5.5 were determined empirically. Hence, testing was performed on each of the eight video clips using a combination of different values for both parameters, where $\epsilon \in [0.55, 0.65, 0.75, 0.85, 0.95]$ and $\delta \in [10, 20, 30, 40, 50]$.

The pair of values that yielded the most number of correct detections (CO) among the eight clips is considered as the best pair and is used in the ALD 2.0. The performance of the ALD 2.0 using the best pair of ϵ and δ is reported in Table 7. We have also provided the performance of ALD 2.0 for all combinations of ϵ and δ that were used in testing in Appendix B. As in Sec. 2.4.9, the total of performance measures CO + MA + MD = 100% and FD + TR = 100% for each clip. Also, if no false detections or true rejections were encountered, then FD = TR = 0.

Table 7: Performance of the ALD 2.0 on the 8 clips using $\delta = 50$ and $\epsilon = 0.95$. The number of images in which false detections and true rejections were encountered are provided in the parenthesis of their columns.

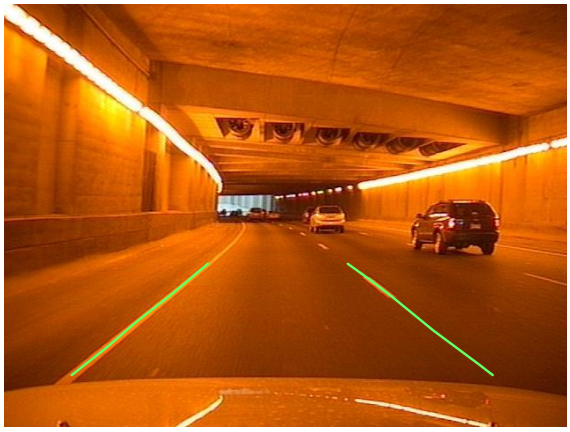
Name	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
Clip 1	90.37	9.20	0.43	100.00 (61)	0.00 (0)	1.08
Clip 2	88.94	10.63	0.43	0.00 (0)	0.00 (0)	2.14
Clip 3	90.63	8.68	0.69	0.00 (0)	0.00 (0)	1.63
Clip 4	97.97	1.94	0.09	0.00 (0)	0.00 (0)	0.56
Clip 5	94.72	5.09	0.19	0.00 (0)	0.00 (0)	1.11
Clip 6	89.91	9.75	0.34	0.00 (0)	0.00 (0)	1.35
Clip 7	99.56	0.44	0.00	0.00 (0)	0.00 (0)	0.67
Clip 8	91.07	8.93	0.00	100.00 (65)	0.00 (0)	0.81

Fig. 109 illustrates the lane boundaries estimated by the ALD 2.0. The lane boundaries are represented by green curves. Unlike the ALD 1.0, the ALD 2.0 has no restrictions regarding illumination conditions and detects the lane boundaries on city roads and highways, and also in the presence of traffic and changing illumination. Again, the error E(n) (details in Sec. 2.4.8) is only computed for the images in which a correct detection occurred. Furthermore, the histogram of E(n) for each clip is also provided in Appendix B.

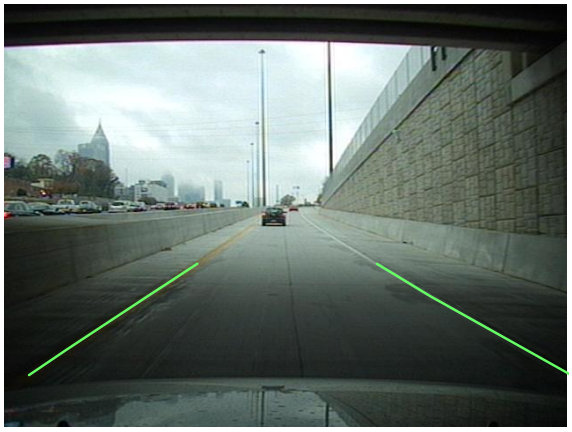
As with the ALD 1.0, the False Detection (FD) values in Table 7 are also high; therefore, in the parenthesis, we have provided the number images in which false detections occur to show that this number is small compared to the number of images in the clip. A few instances of missed detections (MD) and misalignments (MA)

(a) Highway with traffic.

(b) Changing illumination.



(c) Inside a tunnel.



(d) Exiting a tunnel on an access road.

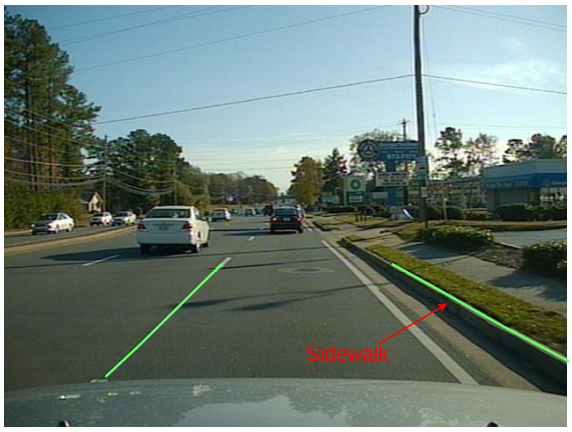
(e) Night time on highway.

(f) Local street at night.

Figure 109: Examples of correct lane detections.

are shown in Fig. 110. Two common causes of misalignments were lens flares and sidewalks. A lens flare appears as a lane marker in the world image as shown in Fig.

(a) Lens flare causes misalignment.



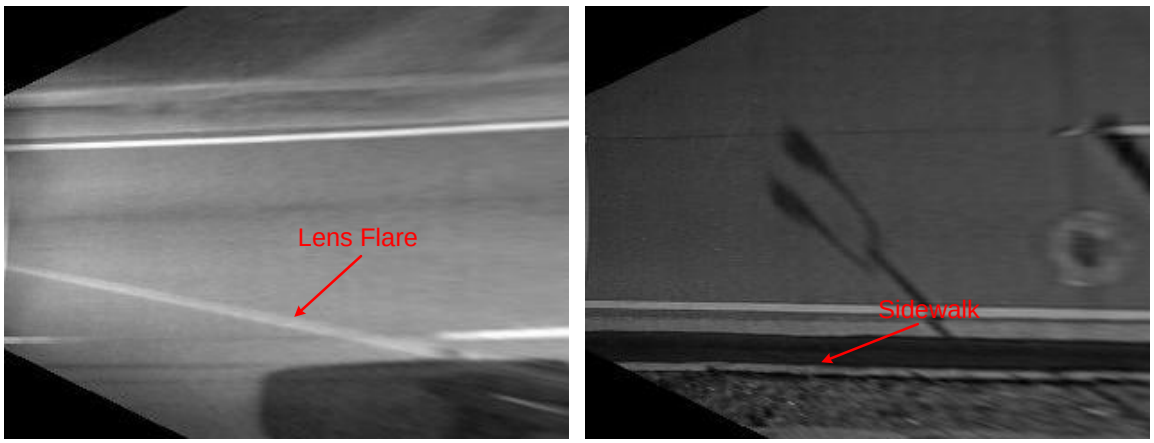
(b) Sidewalk detected on the right.

(c) Misalignment occasionally caused by traffic in neighboring lanes.

(d) Worn out markers cause missed detections.

Figure 110: Examples of misalignments and missed detections.

111a; as a result, it is detected and ultimately leads to a misalignment as shown in Fig. 110a. Similarly, sidewalks also appear as lane markers in the world image as shown in Fig. 111b; therefore, they are detected on few occasions as shown in Fig. 110b. Besides lens flares, traffic in neighboring lanes also contributed to the occasional misalignments as shown in Fig. 110c. The most common cause of a missed detection was the absence of lane markers often due to age and wear on local city roads as shown in Fig. 110d.



(a) Lens flare appears as a diagonal lane marker. (b) Portion of the sidewalk appears as a lane marker.

Figure 111: Causes for misalignments examined the world image.

2.5.8 Closing Remarks

In this section, we presented the ALD 2.0, a new methodology for lane detection. The methodology begins with IPM [20] where the camera image is converted to a bird's-eye view followed by a color conversion from RGB to YCbCr. Next, in Filtering, Template matching and Hysteresis threshold are used to create a number of binary images in which lane markers are detected. Then, using the Lane Region Merging block, the binary images provided by the previous block are combined into a single binary image in which lane markers are visible with many artifacts removed. This is followed by a Lane Marker Classifier that eliminates artifacts that may be present in the binary image by subjecting the detected lane markers to angular and window constraints. Finally, the Piecewise Tracker uses the detected lane markers to estimate the position of the lane boundary from one image to the next. The ALD 2.0 is also implemented in Matlab and operates at about $0.8 - 1$ fps on the test bed computer. With code optimization, implementation on dedicated hardware, and possible Graphics Processing Unit (GPU) based multi-core acceleration, a real-time system could be realized.

Based on the performance evaluation, we observed that the ALD 2.0 was able to

yield good rates for correctly detecting lane markers with small errors on city roads and highways, in the presence of traffic, various illumination and shadow conditions. Unlike the ALD 1.0, it is not limited to operating under any particular illumination condition. However, in cases as shown in Fig 110, lens flares and sidewalks cause misalignments. This is because the current system detects left and right lane boundaries independently. By detecting the lane boundaries as a pair of lines or curve that are subject to certain distance and angle constraints, these misalignments can be reduced. This is type of line pairing is the basis of the new lane detector that will be introduced in the next section.

2.6 Advanced Lane Detector 3.0

In this section, we introduce our final lane detection methodology that is called the ALD 3.0. The ALD 3.0 focuses on finding the lane marker features that are not only parallel, but also meet certain distance and angle requirements between each other. It is an improvement over the ALD 2.0 and its layout is shown in Fig. 112. First, the images captured by the camera undergo a conversion from color to grayscale followed by geometric transformation that is implemented through Inverse Perspective Mapping [20]. Then, in the Filtering block, the transformed images undergo Template matching, followed by masking, and thresholding to create a binary image in which

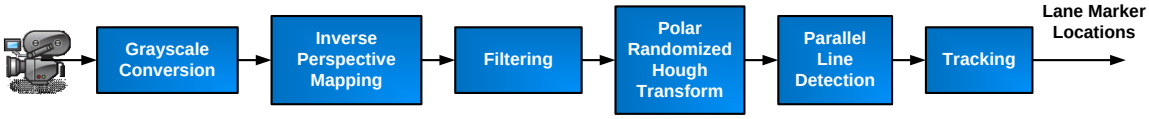


Figure 112: Layout of the ALD 3.0.

the lane markers are detected. A new technique called the Polar Randomized Hough Transform (PRHT) [41] is introduced and is used to efficiently find lines in an image. Then, in the Parallel Line Detection block, the detected lines are paired if they meet certain distance and angular requirements between each other. Finally, the paired lines are tracked from one image to the next using a tracker. The ALD 3.0 is tested with videos that reflect real world driving conditions and its performance is discussed later in the section.

Although the ALD 2.0 demonstrated good performance as shown in Table 7, it was prone to misalignments caused by lens flares. Some of these misalignments are shown in Fig. 110. These misalignment were often caused by the independent detection of the left and the right lane markers. As a remedy, the Parallel Line Detection block that is introduced in this section focuses on selecting the lane marker features in pairs. Furthermore, the selected pairs must meet certain distance and

angle requirements between each other which help reduce the detection of artifacts and causes for misalignments. Additionally, the ALD 2.0 was slow, operating at less than 1 fps in most cases. Hence, improving the execution time of lane detection is a necessary requirement. With the ALD 3.0, we introduce several enhancements and optimization techniques that help meet this requirement. These techniques will be covered in detail below.

2.6.1 Grayscale Conversion

In this block, the input RGB color image $I(n)$ is converted to grayscale. The Grayscale transformation [25] is performed as

$$\hat{I}_n(r,c) = 0.299 \cdot R_n(r,c) + 0.587 \cdot G_n(r,c) + 0.114 \cdot B_n(r,c) \tag{54}$$

where $R_n(r,c)$, $G_n(r,c)$, and $B_n(r,c)$ are the red, green, and blue values of the pixel at location (r,c) in $I(n)$ and $\hat{I}_n(r,c)$ is the resultant intensity value of the pixel at (r,c) in the grayscale image $\hat{I}_n$. $I(n)$ and $\hat{I}_n$ are shown in Fig. 113. This transformation

(a) Color image $I(n)$. (b) Grayscale image $\hat{I}_n$.

Figure 113: Detection of the white and the yellow lane markers in grayscale images.

is identical to one in Eq. (2) except here R_n, G_n, and B_n belong to a single color image $I(n)$ while in Eq. (2), R_n, G_n, and B_n belong to the Average Image $\bar{I}_K(n)$. Although all three color channels were used by the ALD 2.0 in the detection of white

and yellow markers, it can be observed in Fig. 114 that during both day and night, both white and yellow markers appear bright and are clearly visible in the grayscale (or the Y channel) image. Hence, detection of the white and the yellow lane markers

(a) Day. (b) Night.

Figure 114: The white and the yellow lane markers in grayscale images.

in the ALD 3.0 is performed using only the grayscale image. This conversion helps reduce the execution time of the lane detector since it is only necessary to process a single channel of the image.

2.6.2 Inverse Perspective Mapping

IPM has been explained previously in this dissertation; therefore, please refer to Sec. 2.4.2 for more details. The application of IPM converts the grayscale camera image $\hat{I}_n$ to the world image W_n as shown in Fig. 115.

2.6.3 Filtering

The focus of this block is to convert the grayscale world image to binary in which lane markers appear as thin line segments [42]. As shown in Fig. 116, this block consists of three stages:

1. Template matching

2. Masking

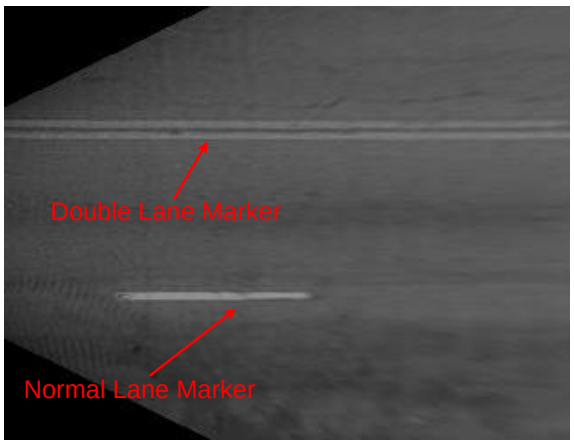


Figure 115: $\hat{I}_n$ converted to the world image W_n.

3. Thresholding.

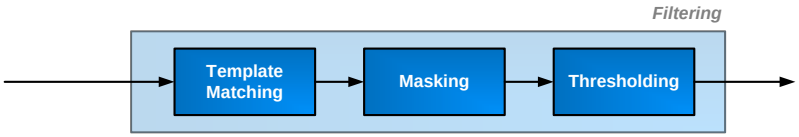
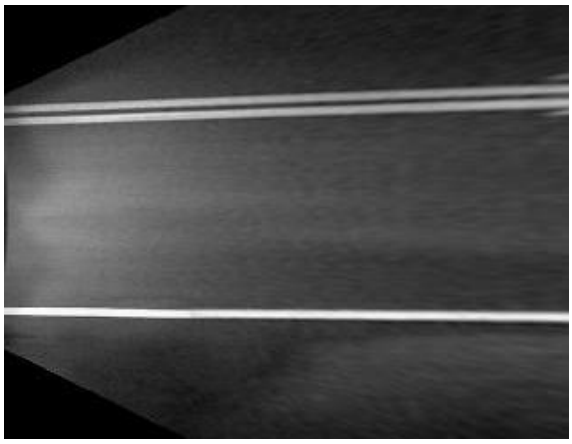


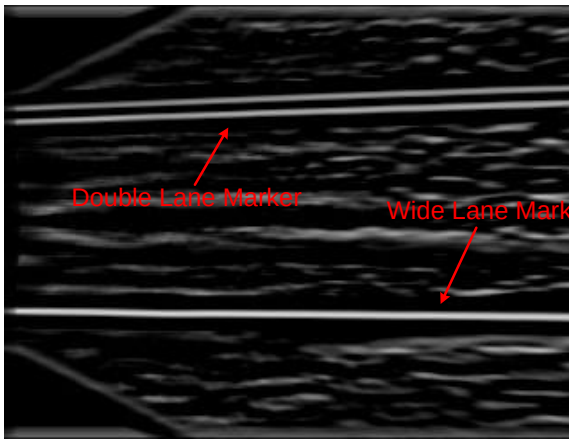
Figure 116: Stages in the filtering block.

First, Template matching is performed using Normalized Cross Correlation [30] as described in Sec. 2.5.3. However, unlike the ALD 2.0, where template matching is performed on all three color channels of world image; here, template matching is only performed on a single channel. Additionally, only the template of the normal lane marker that is shown in Fig. 69 is used in template matching. We only use the template of the normal lane marker because we empirically determined it to be the most versatile of the three. To elaborate, we observed that the correlation between world images and the template of the normal lane marker produced coefficient maps with large coefficients in areas that correspond to normal, wide, and double lane markers. Whereas images that were correlated with templates of either the wide or double lane marker produced coefficient maps with large coefficients only in the areas corresponding to their respective templates. An example of this observation is shown in Fig. 117 where we have provided a world image along with the coefficient maps

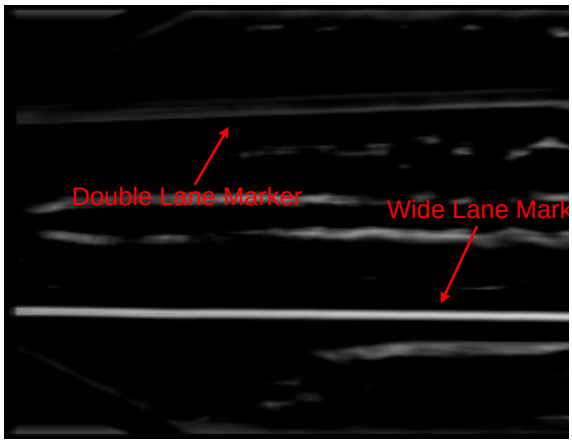
that were created by as a result of correlation between the image and each of the three templates. As a result, a single template can be used to locate all three lane



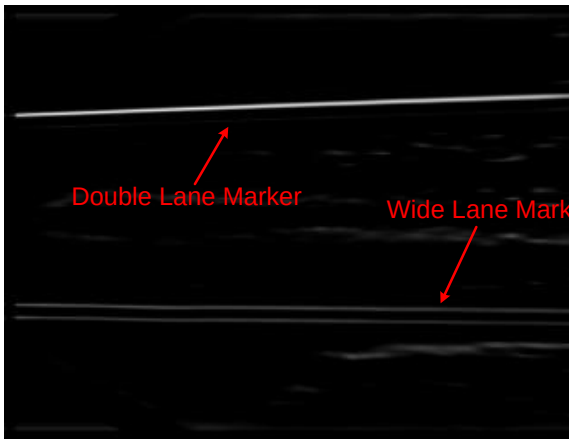
(a) Image containing the double and the wide lane markers.



(b) Coefficients map from correlation with the normal lane marker template.



(c) Coefficients map from correlation with the wide lane marker template.



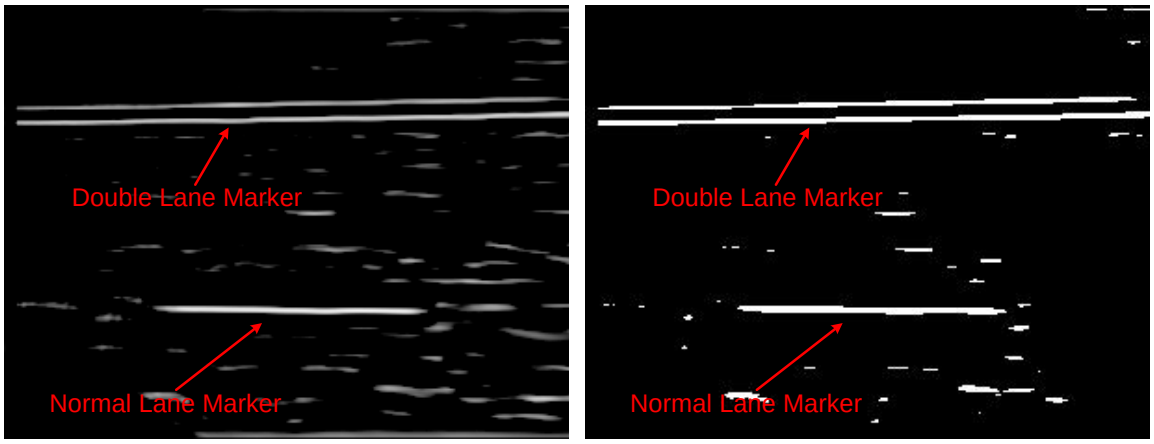
(d) Coefficients map from correlation with the double lane marker template.

Figure 117: Coefficients maps created by correlation with the different templates.

markers in the image. Consequently, Template matching described here is computed between the world image W_n and the template of a normal lane marker. The result is a single coefficients map C_n in which all three lane markers are highlighted. By comparison, Template matching in the ALD 2.0 results in nine coefficient maps as discussed in Sec. 2.5.3.

Next, in masking, the idea is to ignore pixels in the image that do not correspond to the lane markers. This is done by applying a binary mask to W_n. Furthermore,

this mask is created by application of the Hysteresis threshold to the coefficients map C_n. For details on Hysteresis threshold, please refer to Sec. 2.5.3. The value for $\tau_{Hi} = 0.609$ is based on the cut off point of the Receiver Operator Characteristic (ROC) curve described in Sec. 2.5.3. The value for $\tau_{Lo} = 0.578$ is based on the threshold multiplier ϵ that yielded the best result in the performance evaluation of the ALD 2.0 from Sec. 2.5.7. The result of hysteresis threshold is a binary image $\hat{B}_n$ in which pixels mostly corresponding to lane markers are detected as shown in Fig. 118. This binary image is the mask that is applied to W_n. The creation of the binary



(a) Grayscale coefficients map C_n .

(b) Binary mask $\hat{B}_n$.

Figure 118: Converting the coefficients map to binary using Hysteresis threshold.

mask $\hat{B}_n$ can be summarized as

$$C_n \geq_H (\tau_{Hi}, \tau_{Lo}) = \hat{B}_n \tag{55}$$

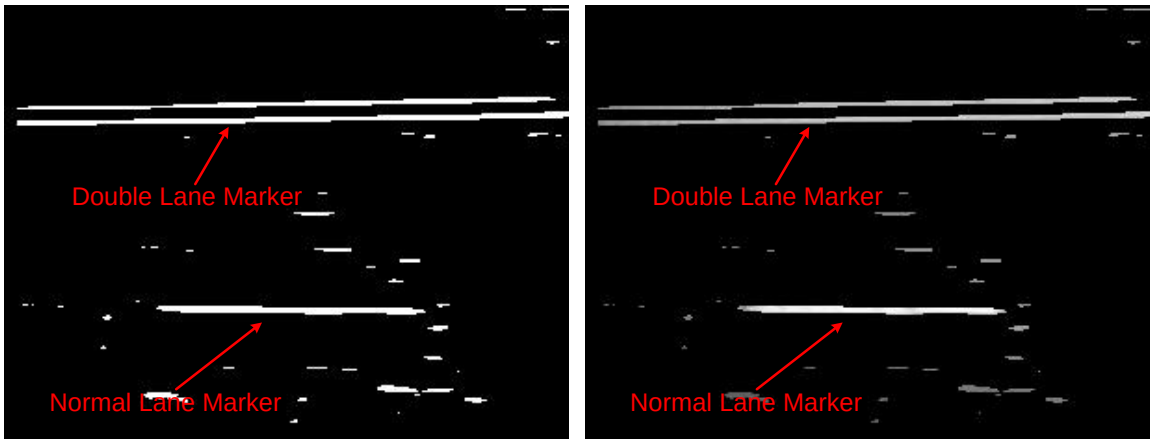
where $\geq_H$ is the symbol for the Hysteresis threshold. Then, $\hat{B}_n$ is multiplied pixel-by-pixel with W_n resulting in $\hat{W}_n$,

$$\hat{B}_n \cdot W_n = \hat{W}_n \tag{56}$$

where each pixel (x, y) in $\hat{W}_n$ is computed as

$$\hat{W}_n(x,y) = \begin{cases} 0, & \text{if } \hat{B}_n(x,y) = 0 \\ W_n(x,y), & \text{if } \hat{B}_n(x,y) = 1 \end{cases} \tag{57}$$

As a result, $\hat{W}_n$ is a masked copy of W_n in which only the raw pixels intensities that correspond to lane markers are preserved. $\hat{W}_n$ is shown in Fig. 119b. It should be noted that $\hat{B}_n$ may contain some artifacts as shown in Fig. 119a. These artifacts



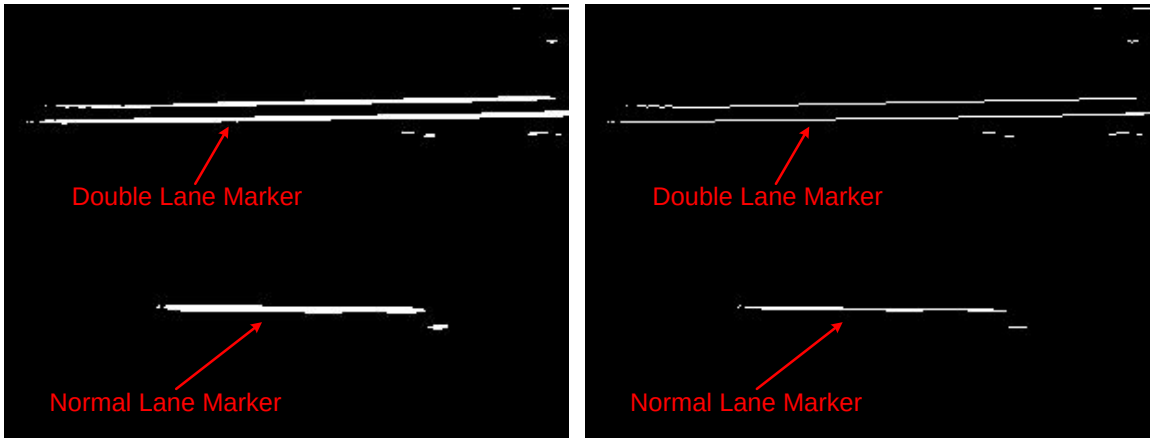
(a) Binary mask $\hat{B}_n$. (b) $\hat{W}_n$, the masked copy of W_n.

Figure 119: Masking pixels in W_n.

often correspond to stray objects or portions of the road surface. As a result, $\hat{W}_n$ may also contain some pixels that belong to the artifacts.

Lastly, in Thresholding, $\hat{W}_n$ is converted to a binary image B_n in which lane markers appear as thin line segments [41]. $\hat{W}_n$ is primarily a black image and contains a few non zero pixels. These non zero pixels can correspond either to the lane markers or the artifacts as shown earlier. However, we observed that the bright, non-zero pixels in $\hat{W}_n$ often correspond to either white or yellow lane markers. On the other hand, the dark, non-zero pixels often correspond to the artifacts in $\hat{W}_n$. Therefore, the lane markers are detected by applying a threshold α to $\hat{W}_n$ such that only the bright pixels from $\hat{W}_n$ are preserved and the dark pixels are suppressed. The value of α is computed as the β^{th} percentile of the Cumulative Density Function (CDF) of all non-zero pixel intensities in $\hat{W}_n$. Since α is computed based on the global statistics of $\hat{W}_n$, it is not a static value. α changes from one image to the next. Furthermore, we were not able to determine the ideal value for β analytically; hence, we tested β with several empirically determined values and selected the one that produced the

best result. The different values of β that were used in testing are listed in Sec. 2.6.7. The application of the threshold α to each pixel in $\hat{W}_n$ results in a binary image $\tilde{B}_n$ in which the lane markers are detected. Finally, $\tilde{B}_n$ is morphologically thinned to create B_n. The reason for thinning is because the lane markers appear as thick objects in $\tilde{B}_n$. By application of thinning, these thick objects can be converted to thin lines and thus meeting the requirements of this block. $\tilde{B}_n$ and B_n are shown in Fig. 120.



(a) $\tilde{B}_n$. (b) $\tilde{B}_n$ is morphologically thinned to create B_n.

Figure 120: Detecting the lane markers in the binary image.

Besides computing the β^{th} percentile of the CDF, other approaches such as computing the trimmed mean, trimmed median, and local statistic based methods could be used to determine α and produce a similar binary image $\tilde{B}_n$. The Filtering block described above is much faster than the same block in ALD 2.0 for several reasons:

1. Template matching is performed on only one coefficient map rather than nine coefficient maps in the ALD 2.0.

2. Hysteresis threshold is used to create a single binary mask $\hat{B}_n$ rather than nine masks in the ALD 2.0.

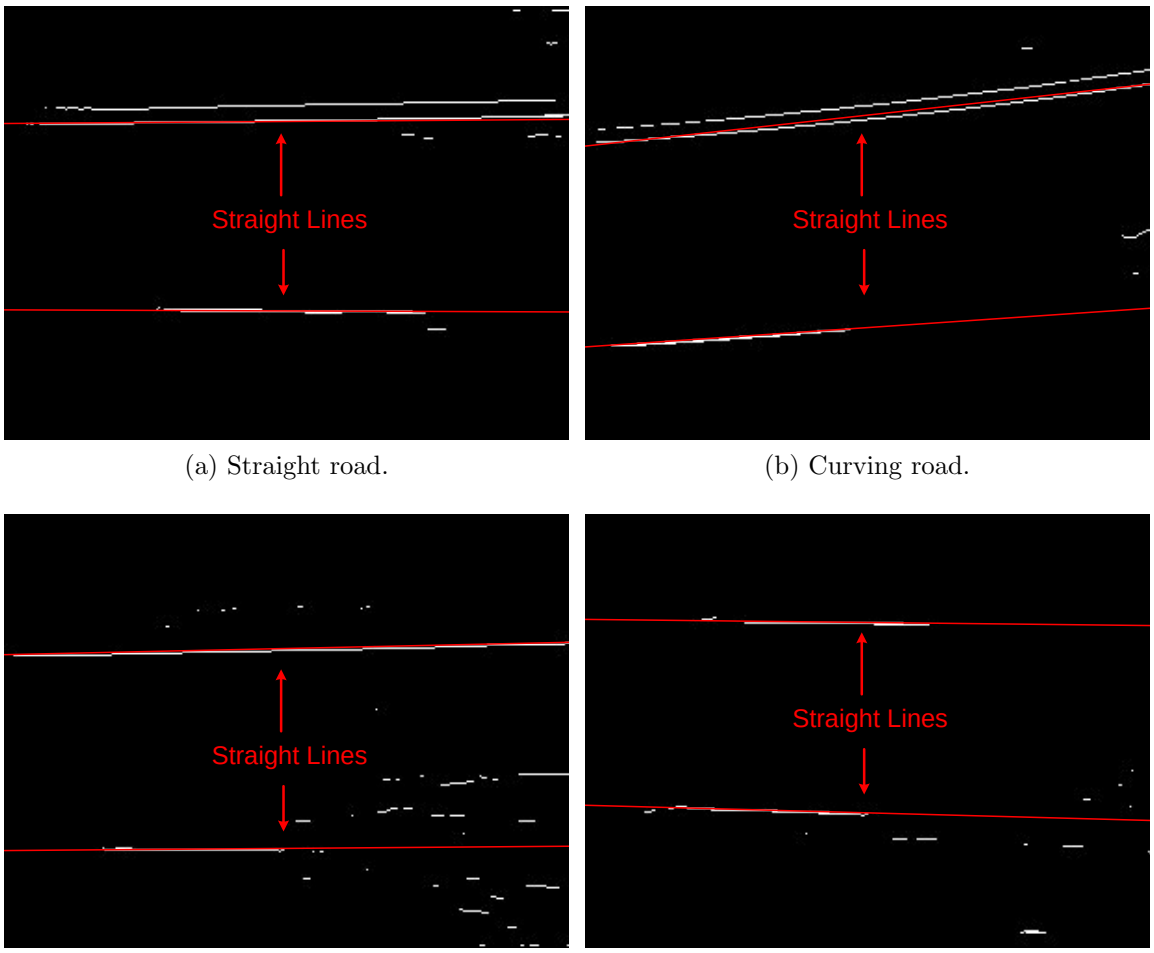
3. The output of the Filtering block is a single binary image in which lane markers are detected. This is not the case with the ALD 2.0 as the output of the Filtering block still needs to undergo further processing.

As a result, it is clear that significantly more computation is needed in the ALD 2.0 prior to creating a single binary image in which lane markers are detected. These are some of the factors that contribute to its slow execution time.

2.6.4 Polar Randomized Hough Transform

The output of the Filtering block is a binary image B_n in which lane markers appear as thin lines. However, if we observe B_n closely, we can see that a dashed lane marker appears as a short line segment in the image. Furthermore, a solid lane marker on a straight road has the appearance of a straight line. And on curving roads, a solid lane marker appears as a line that is slightly curved. However, a straight line can be used to provide a good representation of the curve. In Fig. 121, we have shown a few binary images in which the lane markers appear as thin lines. Furthermore, we have also shown the straight lines that are used to represent the "thinned" lane markers in the images. Therefore, detecting straight lines in B_n is analogous to detecting the lane markers in W_n. Moreover, the straight lines represent the lane boundaries on which the lane markers lie. Therefore, the focus of this block is to find a collection of straight lines in the binary image B_n.

Although numerous methods exist, the classical Hough transform [22] is still one of the most common methods that is used to accomplish this task. Details of the Hough transform along with an example can be found in Sec. 2.4.4. The Hough transform generates a Hough space that contains a number of peaks. Moreover, a peak in the Hough space at the location (ρ, θ) corresponds to a line in the image whose perpendicular distance from the origin is ρ, and θ is the angle formed by the perpendicular and the origin. By finding distinct peaks in the Hough space, a number of straight lines can be recovered in the binary image. However, the classical approach tends to be very slow. This is because the Hough space is populated by exhaustively incrementing a range of (ρ, θ) locations for each non-zero pixel in the binary image.



(a) Straight road.

(b) Curving road.

(c) Straight road.

(d) Curving road.

Figure 121: Using straight lines to model lane markers on the road.

To speed up this process, the Low Resolution Hough Transform (LRHT) [29] was introduced in Sec. 2.4.4. In comparison to the classical Hough transform, the LRHT uses a limited sweep interval for θ. In addition, the LRHT also uses larger increments between the θ values. As a result, the LRHT reduces the number of computations required to generate the Hough space. However, the larger increments sacrifice precision of the detected lines for speed. Although the LRHT was used to detect lines in the ALD 1.0, the detected lines were only rough estimates. Each line was subject to further processing in later stages of the ALD 1.0. Therefore, a technique that can find lines quickly and with good precision is needed. One such method is the Randomized Hough Transform.

The Randomized Hough Transform (RHT) operates iteratively by randomly sampling a set of points to compute a single location in the Hough space that is incremented [43]. To elaborate, we start with a set S_N that contains the locations of the N non-zero pixels in the binary image. Then, two pixels at (x_1, y_1) and (x_2, y_2) are randomly selected with replacement from S_N. Since two pixels are trivially collinear, the parameters of the line on which they lie can be determined by solving the system of equations below:

$$y_1 = mx_1 + b \tag{58}$$

$$y_2 = mx_2 + b \tag{59}$$

where m represents the slope and b represents the y-intercept of the line. The recovered parameters are then used to increment the element at index (m, b) in the Hough space. This process of selecting a pair of pixels from S_N, and solving the system of equations is repeated $\binom{N}{2}$ times. One point to note is that the RHT uses a Hough space that is indexed by (m, b), unlike the previously discussed cases where the Hough space was indexed by (ρ, θ). The underlying idea of the RHT is illustrated in Fig. 122 with the two randomly selected non-zero pixels shown in red and the computed location of the element in the Hough space shown in green. The RHT takes advan-

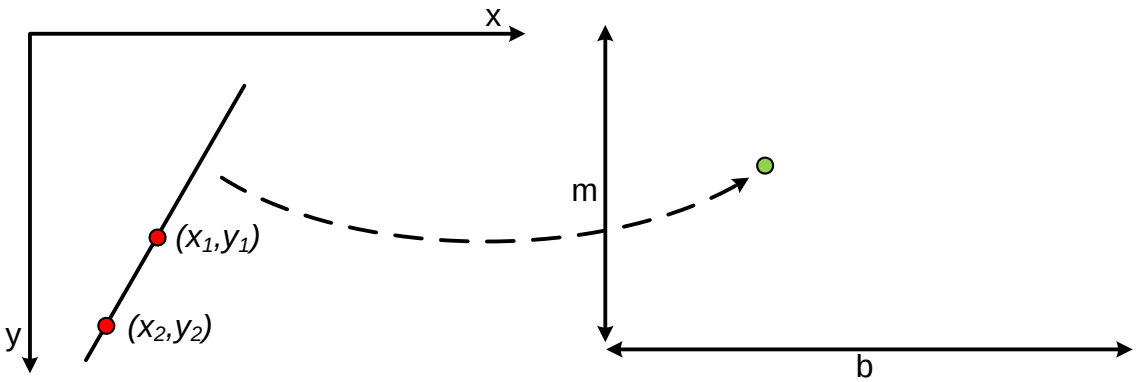


Figure 122: Randomized Hough Transform.

tage of the fact that a straight line can be fully realized using two points, there by avoiding the exhaustive computation used in the classical HT and providing a speed

improvement. However, the RHT has a major shortcoming. Since the RHT recovers parameters using the slope-intercept form, problems arise when the two points lie on a near vertical line. In such a condition, the value of both parameters m and b may be unbounded. As a result, incrementing the appropriate the element in the Hough space becomes difficult. Therefore, to handle this problem, it is beneficial to represent the line in the polar form. This is where the Polar Randomized Hough Transform (PRHT) comes in [41].

Similar to the RHT, PRHT also operates iteratively and uses a random sampling of points. At first, two non-zero pixels $\mathbf{P}_1$ and $\mathbf{P}_2$ are randomly selected with replacement. The location of each pixel is written in vector notation as shown below:

$$\mathbf{P}^T = [x \ y] \tag{60}$$

Consider the origin $\mathbf{O}$ at the top left of the image. The projection of the origin $\mathbf{P}_o$ on the line formed by $\mathbf{P}_1$ and $\mathbf{P}_2$ is computed as

$$c = \frac{(\mathbf{O} - \mathbf{P}_1)^T \cdot (\mathbf{P}_2 - \mathbf{P}_1)}{\|\mathbf{P}_2 - \mathbf{P}_1\|^2} \tag{61}$$

$$\mathbf{P}_o = \mathbf{P}_1 + c \cdot (\mathbf{P}_2 - \mathbf{P}_1) \tag{62}$$

$\mathbf{OP}_o$ is the normal drawn from the origin to the line. The angle between the x-axis and $\mathbf{OP}_o$ represents θ in the normal form of a line equation and is calculated as

$$\theta = \arctan\left(\frac{y_o}{x_o}\right) \tag{63}$$

Similarly, ρ in the line equation is calculated as

$$\rho = x_o \cdot \cos\theta + y_o \cdot \sin\theta \tag{64}$$

where x_o and y_o are x and y co-ordinates, respectively, of $\mathbf{P}_o$. Using Eq. (63) - (64), the element at (ρ, θ) in the Hough space is incremented. On the next iteration, another pair of pixels is randomly selected with replacement and Eq. (61) - (63) are repeated.

The concept of the PRHT is illustrated in Fig. 123 with the two randomly selected pixels shown in red, the projection of the origin shown in blue, and the computed location of the element in the Hough space shown in green. The Polar Randomized

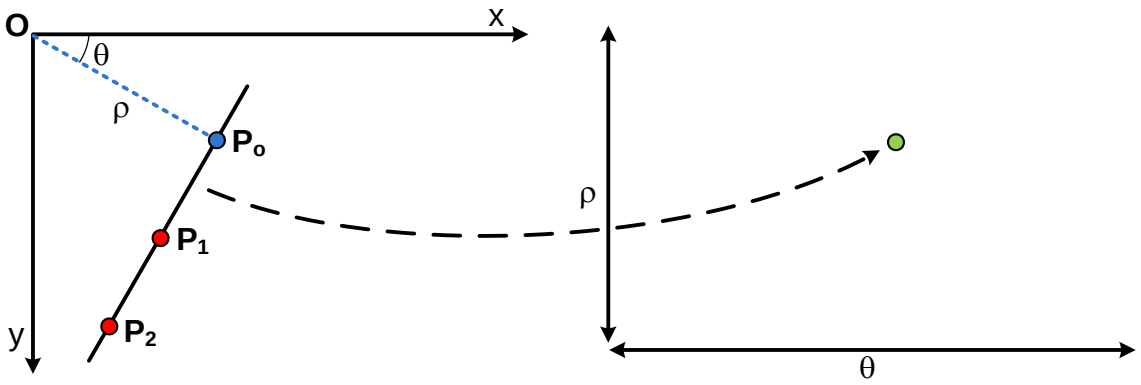
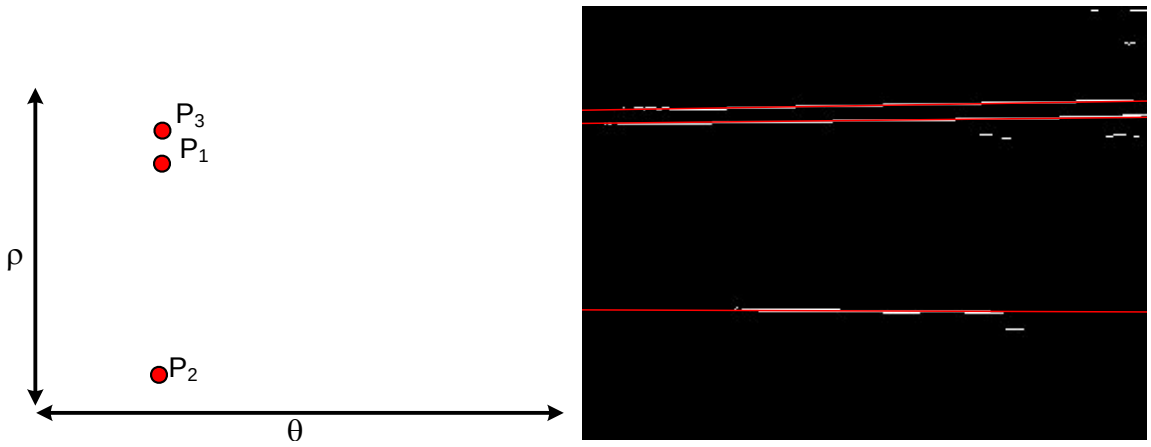


Figure 123: Polar Randomized Hough Transform.

Hough Transform has the benefits of both classical and Randomized Hough Transform. By using polar descriptors (ρ, θ), a line can be extracted and described without restriction on its orientation. And, with random sampling of points, the speed benefit of the Randomized Hough Transform is also attained [41]. Similar PRHT computations are shown in [44, 45]. However, in [44], the parameters of the line are first recovered in terms of the slope-intercept parameters (m, b) and then converted to the polar descriptors (ρ, θ). In the PRHT, no intermediate computation of the slope-intercept parameters is needed. Then, the approach shown in [45] uses three collinear points to compute the (ρ, θ) parameters while the PRHT only use two.

Finally, we conclude this block by using the PRHT to find at most 10 best fitting lines in B_n. These 10 lines correspond to the 10 highest peaks in the Hough space. Furthermore, each peak must exceed a value of 50 so that at least 50 non-zero pixels lie on each detected line in B_n as described previously in Sec. 2.4.4. In this particular example, only 3 peaks exceed 50 in the Hough space. These peaks are labeled and shown in Fig. 124a. As a result, only three lines are detected in B_n. The Hough space and the detected lines are shown in Fig. 124



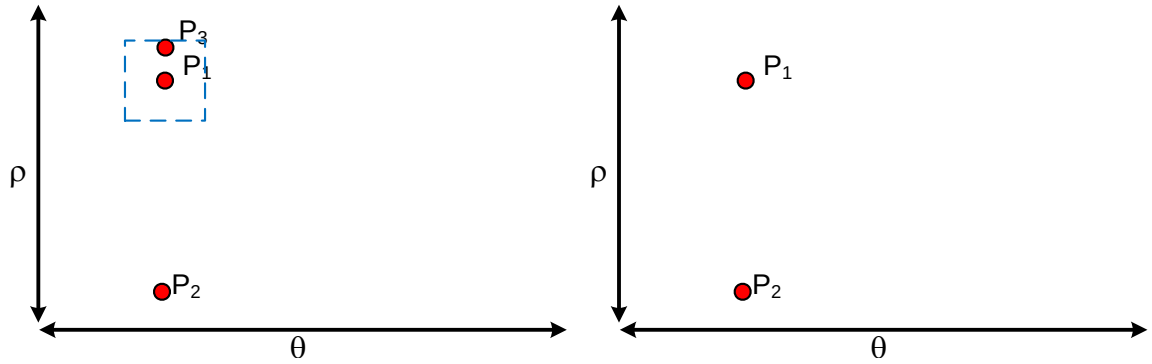
(a) Hough space with labeled peaks. (b) Lines detected in B_n.

Figure 124: Detected lines and corresponding Hough space.

Finally, if two peaks are near each other, then the smaller peak is merged into the larger peak. For example, in Fig. 125a, P_3 is considered to be near P_1 since P_3 lies inside the dotted blue square centered at P_1. Each side of the square has an empirically determined length of 2 units. This implies that difference in ρ values between the two peaks is 2 ft, and the difference in θ values between the two peaks is 2°. In addition, the value of P_1 is more than the value of P_3. Therefore, P_3 is merged with P_1. As a result, the Hough space now only contains two peaks as shown in Fig. 125b. Consequently, B_n now contains two lines instead of three as shown in Fig. 125c.

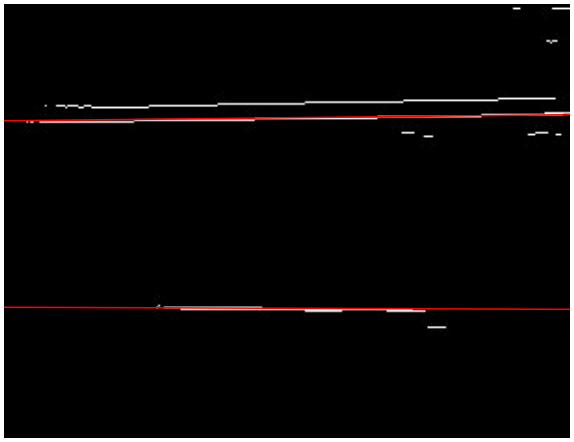
2.6.5 Parallel Line Detection

The focus of this block is to select the ideal pair of lines from B_n to represent the lane markers. The choice is made by subjecting each pair of lines to certain distance and angular requirements between each other [42]. As mentioned in the previous block, the 10 best fitting lines in the binary image B_n correspond to the 10 highest scoring peaks in Hough space. Furthermore, each peak needs to exceed a value of 50. The Hough space from the previous block contained only three peaks that met these requirements. After merging, it contains only two peaks. To determine if two lines are



(a) P_1 and P_3 are near each other.

(b) P_3 is merged into P_1.



(c) B_n now contains only two lines.

Figure 125: Merging peaks and detecting lines.

parallel, the relationship between their corresponding peaks needs to be determined. By definition, parallel lines are non-intersecting; as a result, peaks with identical θ values could be paired. Ideally, lane boundaries also run parallel on most roadways; however, due to imperfections in the captured image, camera lens, variations in lane marker placement, they may not appear parallel. As a result, the constraint on requiring identical θ values between a pair of peaks is "loosened" by allowing some tolerance. This tolerance is computed as

$$\Delta\theta = |\theta_1 - \theta_2| \tag{65}$$

where θ_1 and θ_2 are the θ values of the two peaks being compared. Furthermore

$$\Delta\theta \leq \Delta\theta_{max} \tag{66}$$

where $\Delta\theta_{max}$ is the maximum acceptable difference in θ values between the two peaks. We were unable to determine the ideal value of $\Delta\theta_{max}$ analytically; therefore, we empirically determined a range of values for testing $\Delta\theta_{max}$ and selected the value that produced the best result. The different values of $\Delta\theta_{max}$ that were used in testing are listed in Sec. 2.6.7.

Besides angular constraints, parallel lane boundaries must also abide to certain distance constraints. As a result, the peaks corresponding to the parallel lane markers must be within a distance $\Delta\rho$ of each other which is computed as

$$\Delta\rho = |\rho_1 - \rho_2| \tag{67}$$

where ρ_1 and ρ_2 are the ρ values of the two peaks being compared. Furthermore

$$\Delta\rho_{min} \leq \Delta\rho \leq \Delta\rho_{max} \tag{68}$$

where $\Delta\rho_{min}$ is the minimum distance necessary between the two peaks and $\Delta\rho_{max}$ is the maximum acceptable distance between the two peaks. The FHA states that lane boundaries are spaced at least 12 ft apart [19]. As a result, we set $\Delta\rho_{min} = 88$ pixels (the equivalent of 11 ft in the world image, and just under the 12 ft requirement). Since the spacing between lane boundaries varies on different road conditions, we were unable to determine the ideal value of $\Delta\rho_{max}$ analytically; therefore, we empirically determined a range of values for testing $\Delta\rho_{max}$ and selected the value that produced the best result. The different values of $\Delta\rho_{max}$ that were used in testing are listed in Sec. 2.6.7.

The next step is to approximate the center of the lane. We start by drawing a horizontal line that splits W_n through the middle as shown in Fig. 126. We assume this line represents the center of the lane. Furthermore, this line maps to a single point (ρ_l, θ_l) in the Hough space as shown in Fig. 126. Although the vehicle is expected to change lanes during a normal commute, it is safe to assume that during the majority of the commute, the vehicle travels within the lane. However, the vehicle may not

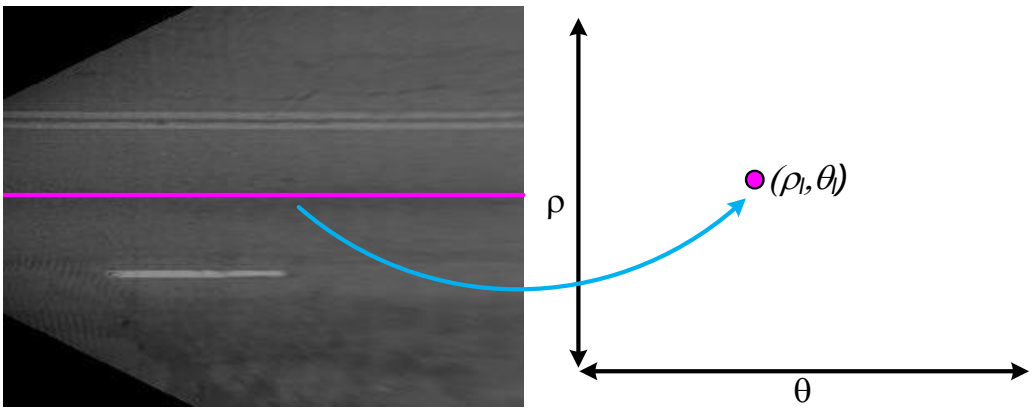


Figure 126: Line splits the world image horizontally.

always be in the center of the lane. Therefore, in W_n, the line representing the lane center will move up or down as shown in Fig. 127 depending on the position of the vehicle in the lane. This results in (ρ_l, θ_l) moving up or down along the ρ axis. On

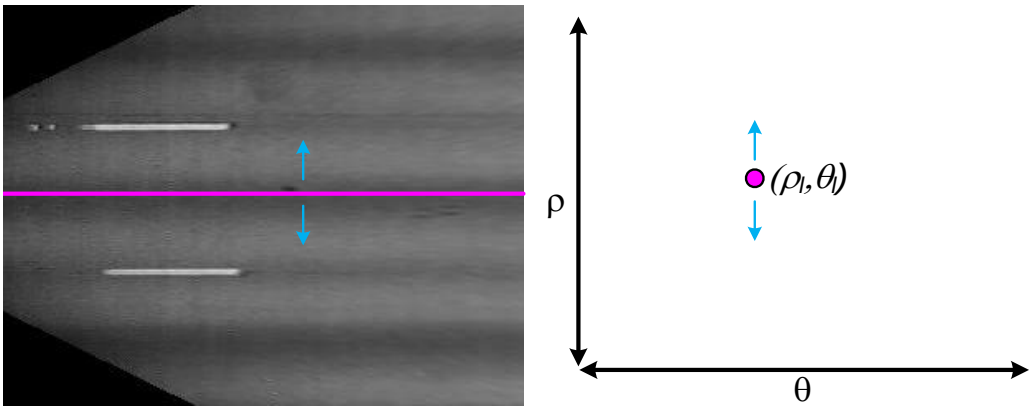


Figure 127: Lane center line moves vertical depending on the position of the vehicle.

curving roads, the angle of this line will change as shown in Fig. 128; as a result, the (ρ_l, θ_l) will move left or right along the θ axis. Therefore, we empirically determined a window that contained within it (ρ_l, θ_l) for most road conditions. This window is shown by the orange dotted rectangle in Fig. 129 and has a height of 64 pixels (equivalent to 8 ft) and a width of 20 pixels (equivalent to 20°). For simplicity, we assume that (ρ_c, θ_c) is the midpoint of the orange rectangle in Fig. 129.

To determine the ideal lane boundary pair, the corresponding peaks in the Hough space (e.g. P_1, P_2) must not only observe both distance and angular constraints, their

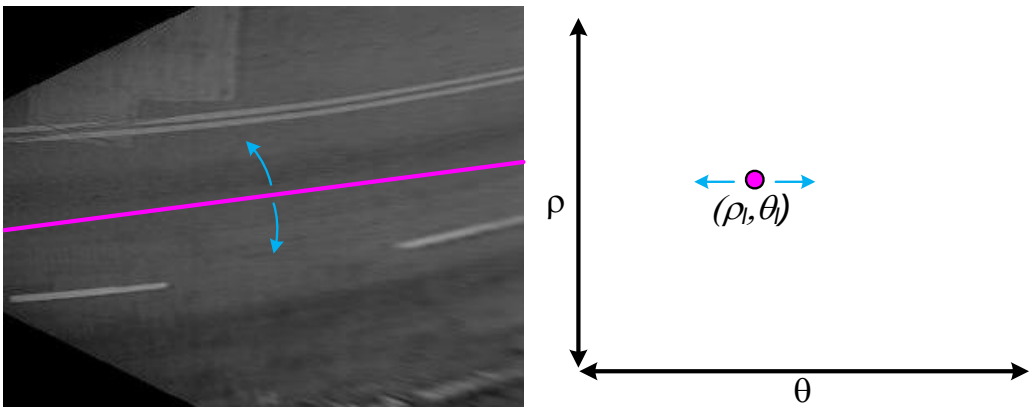


Figure 128: Angle of the lane center line changes on curving roads.

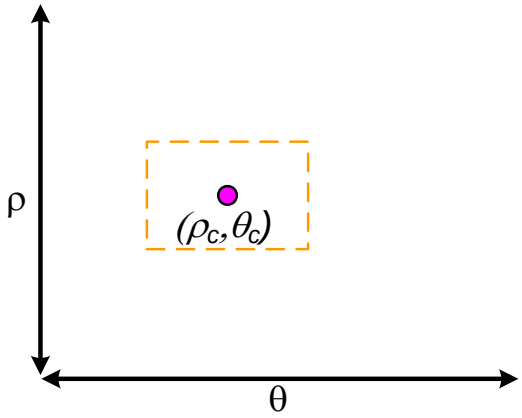
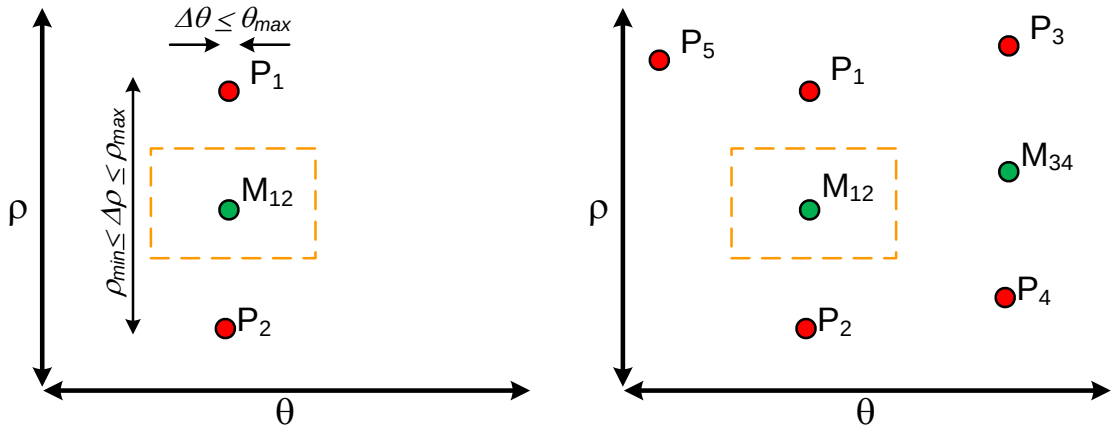


Figure 129: Window that contains (ρ_l, θ_l) for most road conditions. (ρ_c, θ_c) is the center of the rectangle.

midpoint (M_{12}) must also lie within the orange window. This midpoint is shown as a green circle in Fig. 130a. By imposing this restriction on the midpoint, the ideal lane boundary pair is forced to lie near the road center, and, as a result, each lane boundary lies on either side of the vehicle. This restriction on the midpoint also prevents the detection of parallel lines that may be located in other distant parts of the image [42]. For illustration, let us assume that the Hough space contains more than two peaks as shown in Fig. 130b. In this Hough space, both P_3 and P_4 observe the required distance and angular constraints; however, their midpoint (M_{34}) is outside the orange window of the lane center. This suggests that the parallel lines corresponding to these peaks probably belong to the lane boundaries of the adjacent lane or other artifacts



(a) P_1 and P_2 are selected as the ideal pair of lines since their midpoint M_{12} is inside the orange window.

(b) More peaks added to the Hough space for illustration.

Figure 130: Selecting the ideal pair of lines to represent the lane markers.

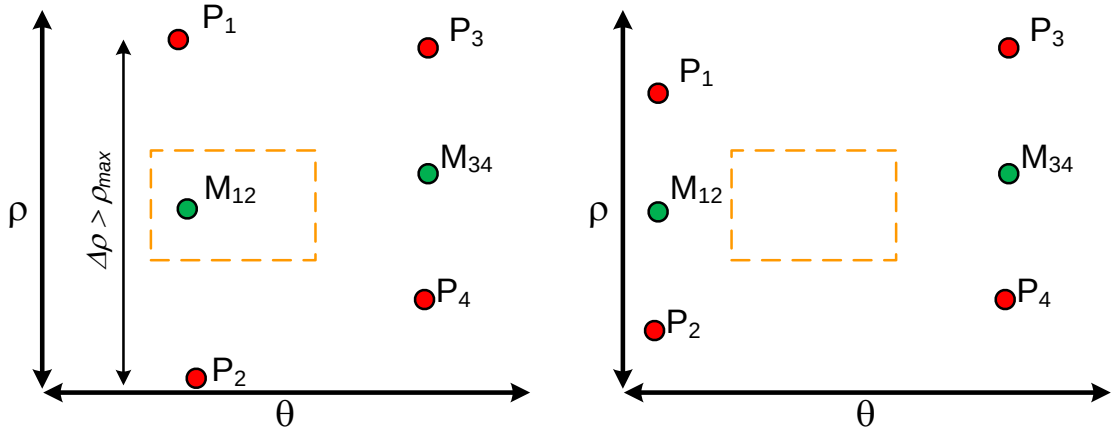
on the road. On the other hand, P_5 does not produce a suitable pair with any peak; therefore, it is ignored. In the situation that P_1 and P_2 are selected as the ideal pair, the line corresponding to P_1 is considered to be the left lane boundary and the line corresponding to P_2 is considered to be the right lane boundary.

Before we conclude this block, we have to consider two special cases:

1. Multiple peaks exist in the Hough space but no pair of peaks meets both, the distance and the angular constraint.

2. Several pairs meet both, the distance and the angular constraints; but, no midpoint lies inside the orange window.

The Hough spaces for the two cases are shown in Fig. 131. In both cases, we only select the highest scoring peak in the Hough space. Then, depending on the ρ value of the peak, the line is determined to be either the left or right lane boundary. If $\rho \leq \rho_c$, then the line is considered to be right lane boundary as shown in Fig. 132. Otherwise, if $\rho > \rho_c$, then the line is considered to be left lane boundary. This is also applies to a Hough space that only contains only one peak that exceeds a value of 50. Lastly, if the Hough space contains no peak that exceeds a value of 50, then this



(a) Multiple peaks but no pair meets both, the distance and the angular constraint.

(b) No midpoint inside the window.

Figure 131: Special cases where there are no appropriate parallel lines.

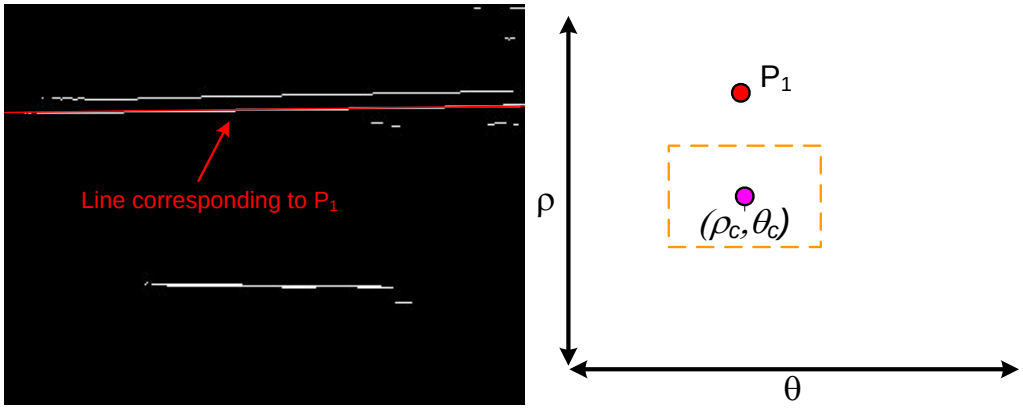


Figure 132: Singe peak corresponds to the left lane boundary since $\rho \leq \rho_c$.

block is skipped entirely.

2.6.6 Tracking

This is the final block of the ALD 3.0 and its purpose is to estimate the position of each line from the selected pair. The estimates are computed using the Kalman filter and are produced in terms of the (ρ, θ) parameters of the lines. Furthermore, the estimates for the left and right lane boundaries are computed separately. The Kalman filter used here is identical to the one in the ALD 1.0; therefore, details regarding its setup, parameters, and estimation in the absence of measurements can be found in Sec. 2.4.7. Finally, the output of this block are the estimates of the left

and the right lane boundaries in the camera image $I(n)$ as shown in Fig. 133.

Figure 133: Estimated lane boundaries after tracking.

2.6.7 Performance Evaluation

In this section, we evaluate the performance of the ALD 3.0. ALD 3.0, as with the previous two ALD's, is also subject to the five performance measures that were described in Sec. 2.4.9. In addition, the ALD 3.0 is tested using the same eight video clips that were used to test both ALD 1.0 and 2.0. If we refer back to the previous blocks of the ALD 3.0, three parameters were determined empirically. These three parameters were: β, a certain percentile of the Cumulative Density Function (CDF) of all non-zero pixel intensities in $\hat{W}_n$ that was described in Sec. 2.6.3; $\Delta\theta_{max}$, the maximum acceptable difference in θ values between a pair of peaks that was described in Sec. 2.6.5, and; $\Delta\rho_{max}$, the maximum acceptable distance between the two peaks was also described in Sec. 2.6.5. Hence, testing was performed on each of the eight video clips using a combination of different values for the three parameters where $\beta \in [0.78, 0.88, 0.98]$, $\Delta\theta_{max} \in [3, 5, 7]$, and $\Delta\rho_{max} \in [104, 120, 136]$. Remember, $\Delta\rho_{max}$ is represented in pixels; therefore, the different values for $\Delta\rho_{max}$ are equivalent to 13 ft, 15 ft, and 17 ft, respectively in the world image. Of all the

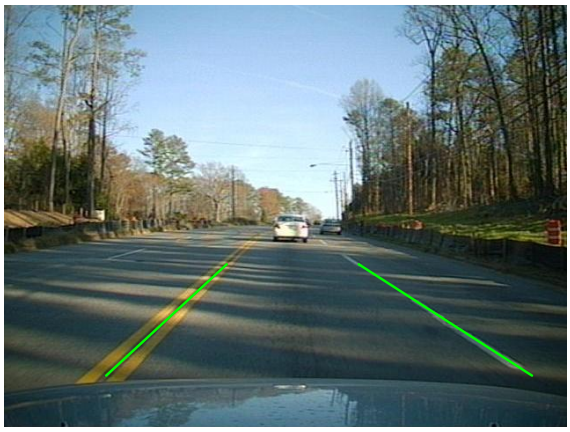
combinations of the three parameters, the triplet that produced the most number of correct detections among the eight clips is considered the best and is used in future implementations of the ALD 3.0. In Table 8, we have reported the performance of the ALD 3.0 that used this triplet. Furthermore, we have also provided the performance of ALD 3.0 for all combinations of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$ that were used in testing in Appendix C. As before, the total of performance measures CO + MA + MD = 100% and FD + TR = 100% for each clip. Also, if no false detections or true rejections were encountered, then FD = TR = 0.

Table 8: Performance of the ALD 3.0 on the 8 clips using $\beta = 0.98$, $\Delta\theta_{max} = 7$, and $\Delta\rho_{max} = 136$. The number of images in which false detections and true rejections were encountered are provided in the parenthesis of their columns.

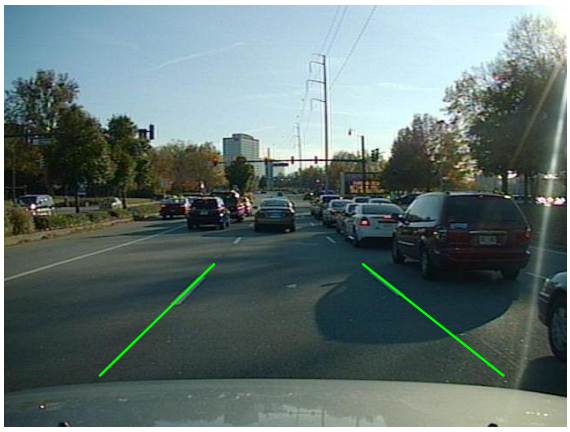
Name	CO %	MA %	MD %	FD %	TR %	$\sigma_{\mathrm{E(n)}}$
Clip 1	88.95	10.62	0.43	100.00 (61)	0.00 (0)	1.36
Clip 2	91.30	8.27	0.43	0.00 (0)	0.00 (0)	2.23
Clip 3	92.46	6.86	0.69	0.00 (0)	0.00 (0)	1.79
Clip 4	98.52	1.39	0.09	0.00 (0)	0.00 (0)	0.57
Clip 5	96.92	2.89	0.19	0.00 (0)	0.00 (0)	1.20
Clip 6	92.97	6.69	0.34	0.00 (0)	0.00 (0)	1.71
Clip 7	99.74	0.26	0.00	0.00 (0)	0.00 (0)	0.69
Clip 8	89.99	10.01	0.00	100.00 (65)	0.00 (0)	0.92

In Fig. 134, we have shown images in which the lane boundaries are estimated by the ALD 3.0. As in the ALD 2.0, the ALD 3.0 is not restricted to operating under any particular illumination conditions. Therefore, it is able to detect lane boundaries during the day and night, in the presence of traffic, and also on city roads and highways. Again, the error E(n) (details in Sec. 2.4.8) is only computed for the images in which a correct detection occurred. Furthermore, the histogram of E(n) for each clip is also provided in Appendix C.

The False Detection (FD) values are high again in Table 8. This problem persists in each of the three lane detectors and is caused by the tracker that is unable distinguish if lane markers are worn out or they are not present on the road. As a result,



(a) Shadow conditions.

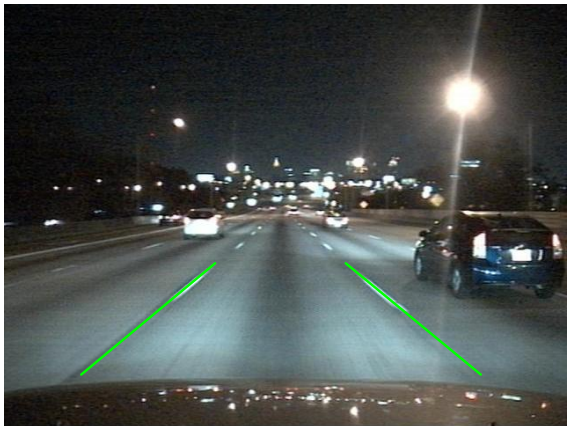


(b) Shadow conditions.

(c) Presence of traffic.

(d) Twilight.

(e) Night time.



(f) Highway at night.

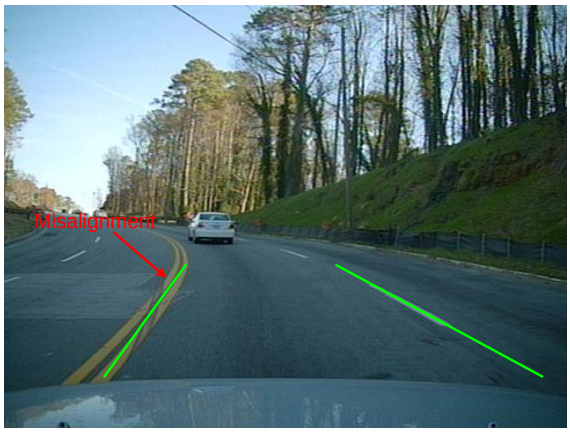
Figure 134: Examples of correct lane detection.

the tracker produces estimates based on prediction for a few images which leads to false detections. Therefore, in the parenthesis we have provided the number images

in which false detections occur to show that this number is small compared to the number of images in the clip.

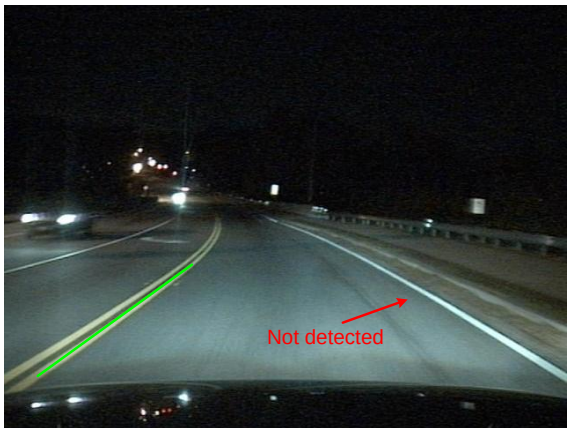
Unlike the ALD 2.0, the ALD 3.0 selects the lane marker features that are parallel. As a result, the false detection of artifacts that were caused by lens flares and neighboring vehicles is greatly reduced. However, the ALD 3.0 on occasion would detect the sidewalk since it also appears as a lane marker in B_n. This shown in Fig. 135a. In addition, the straight line model for lane representation performs well in most cases; even on gentle curving highways and city roads as shown in Fig. 134d and 134e. How-

(a) Sidewalk detected.



(b) Sharp turns may cause misalignments

(c) Worn out makers cause missed detections.



(d) Lane boundaries are spaced more than 17 ft apart.

Figure 135: Examples of misalignments and missed detections.

ever, it has some difficulty on sharp turns. This is illustrated in Fig. 135b which is a

snapshot from Clip 1. On a sharp turn, the straight line model is not able to represent a lane boundary very well. This gives rise to misalignments (MA) and lowers the correct detection rates. As a result, the correct detection rates of the ALD 3.0 are low in comparison to ALD 2.0 for Clip 1. As with both previous ALDs, the absence of the lane markers on local city roads often due to age and wear was a common cause of a missed detection. An example of this situation is shown in Fig. 135c. Missed detections also occurred if the lane boundaries were spaced more than 17 ft apart. In this case, the $\Delta\rho$ between the two peaks exceeds 136 or 17 ft; hence, the peaks are not paired and only the largest peak is selected. An example of this situation is shown in Fig. 135d.

2.6.8 Closing Remarks

We presented a new lane detector in this section that we called the ALD 3.0. The ALD 3.0 performs lane detection by finding a pair of parallel lane marker features that meet certain distance and angle requirements between each other. The ALD 3.0 is made up of a number of blocks and its layout is shown in Fig. 112. First, the captured images undergo a color transformation from RGB to grayscale. Next, the grayscale image is converted to a bird's-eye view using the geometric transformation called Inverse Perspective Mapping [20]. Then, the transformed image undergoes a three stage filtering process that consists of Template matching, Masking, and Thresholding. This results in a binary image in which the lane markers appear as thin line segments. Then, line detection is performed on the binary image using a new and efficient technique called the Polar Randomized Hough Transform (PRHT) [42]. Once the lines are detected, they are paired if possible, based on certain criteria. Finally, the paired lines are tracked from one image to the next using a Kalman filter based tracker. ALD 3.0, as with the other two lane detectors, is also implemented in Matlab; however, it operates at about 4 fps on the test bed computer. As with

the ALD 1.0, with code optimization and implementation on dedicated hardware, a real-time system could be realized.

As in the ALD 2.0, the ALD 3.0 was also able to produce good rates for correctly detecting the lane markers with small errors on city roads and highways, in the presence of traffic, various illumination and shadow conditions. By finding features that were parallel and subject to certain constraints, the misalignments caused by lens flares and the vehicles in neighboring lanes is reduced. The ALD 3.0 is considered to be an improvement over the ALD 2.0 for two reasons:

1. The ALD 3.0 is at least 4 times faster than the ALD 2.0.

2. Even though the ALD 3.0 operates on a single grayscale image using a single lane marker template, it was able to produce higher correct detection rates than the ALD 2.0 in a number of clips that were used in testing.

Despite the good results in the performance evaluation, the ALD 3.0 can be further improved. For example, the central idea of the ALD 3.0 is to detect parallel lines that model the lane markers in the image. This idea could be enhanced to detect parallel curves. One possible way of doing this is by performing a variation of the Template matching procedure along each Sampling Column as shown previously in the ALD 1.0. The result of this could be a better localization of lane markers on both straight and curving roads. Another feature that could be added is curve fitting for lane representation. With curve fitting, a parabolic or quadratic curve could be used to represent the lane boundary over the current method using straight lines. As a result, the ALD 3.0 will most likely show a performance improvement if sharp turns are encountered in the test data e.g. Clip 1. However, as with ALD 2.0, ALD 3.0 also occasionally confuses portions of the sidewalk for a lane marker as discussed in Sec. 2.6.7. Therefore, it may be necessary to include some more feature extraction or post-processing blocks to the original layout of the lane detector that would be used

to suppress the detection of the sidewalk.

2.7 Comparison Between Lane Detectors

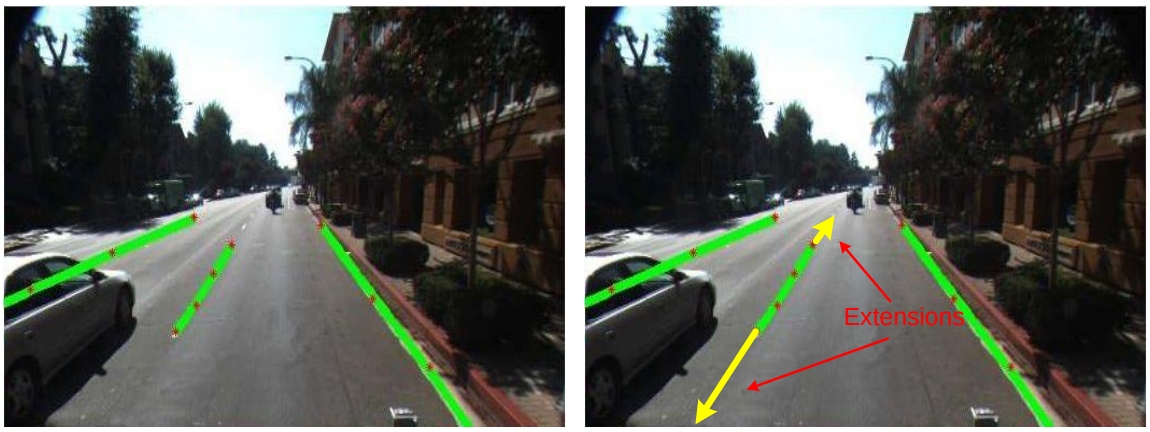
In this section we will compare the performance measures of the three Advanced Lane Detectors (ALD) that were explained in this dissertation. However, since we have already evaluated the ALDs previously, we will not go through the entire performance evaluation again. Rather, we will only list the performance measures of the ALDs for each of the eight clips. By observing the performance measures side-by-side, we will get a better idea of the performance of each lane detector on identical test data. For the reasons discussed in Sec. 2.3, we were unable to determine the standard for comparison among lane detectors; therefore, we implemented the lane detector that was discussed in the related work section. In particular, we implemented the lane detector designed by Aly [3] and reported its performance on the same test data. Besides meeting the requirements and being discussed in the related work section, we implemented this algorithm because:

1. Template matching, IPM, RANSAC, curve fitting procedures that are used by this algorithm have appeared in one or more ALDs in this dissertation.

2. Alice, Team Caltech's entry into the *2007 Darpa Urban Challenge* has used this algorithm to "detect and localize lane lines in urban streets." [21].

3. It has been cited many times by publications in related fields.

This algorithm has been implemented in Matlab and it operates at about $4 - 5$ fps. It should be noted that the original algorithm described by Aly produces estimates that represent only the lane marker and not the boundary of the lane. As a result, a dashed lane marker would appear as a short length segment as shown in Fig. 136a. Therefore, we enhanced the original algorithm by producing a curve to represent the lane boundary. This is done by extending lines from the end points of the curve to and away from the vehicle as shown in Fig. 136b. As with the other lane detectors



(a) Short length segments for dashed lane markers.

(b) Using lines to extend the short length segments.

Figure 136: Extensions using lines.

in this dissertation, the algorithm by Aly [3] is also tested on the eight video clips and is subject to the same five performance measures described in Sec. 2.4.9. The number of images in which false detections and true rejections were encountered are provided in the parenthesis of their columns.

The performance of each lane detector on all eight video clips is provided in the tables below.

Table 9: Comparison between the four lane detectors on Clip 1.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	47.43	51.53	1.04	100.00 (61)	0.00 (0)	1.93
ALD 2.0	90.37	9.20	0.43	100.00 (61)	0.00 (0)	1.08
ALD 3.0	88.95	10.62	0.43	100.00 (61)	0.00 (0)	1.36
Aly [3]	85.49	7.88	6.63	88.80 (55)	11.20 (6)	0.49

Table 10: Comparison between the four lane detectors on Clip 2.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	83.38	15.34	1.28	0.00 (0)	0.00 (0)	0.90
ALD 2.0	88.94	10.63	0.43	0.00 (0)	0.00 (0)	2.14
ALD 3.0	91.30	8.27	0.43	0.00 (0)	0.00 (0)	2.23
Aly [3]	86.96	7.97	5.06	0.00 (0)	0.00 (0)	0.82

Table 11: Comparison between the four lane detectors on Clip 3.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	88.21	10.42	1.37	0.00 (0)	0.00 (0)	0.84
ALD 2.0	88.94	10.63	0.43	0.00 (0)	0.00 (0)	2.14
ALD 3.0	92.46	6.86	0.69	0.00 (0)	0.00 (0)	1.79
Aly [3]	92.33	4.73	2.94	0.00 (0)	0.00 (0)	0.50

Table 12: Comparison between the four lane detectors on Clip 4.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	76.62	21.72	1.66	0.00 (0)	0.00 (0)	0.56
ALD 2.0	97.97	1.94	0.09	0.00 (0)	0.00 (0)	0.56
ALD 3.0	98.52	1.39	0.09	0.00 (0)	0.00 (0)	0.57
Aly [3]	88.90	6.20	4.90	0.00 (0)	0.00 (0)	0.05

Table 13: Comparison between the four lane detectors on Clip 5.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	90.19	7.17	2.64	0.00 (0)	0.00 (0)	0.43
ALD 2.0	94.72	5.09	0.19	0.00 (0)	0.00 (0)	1.11
ALD 3.0	96.92	2.89	0.19	0.00 (0)	0.00 (0)	1.20
Aly [3]	87.85	8.44	3.71	0.00 (0)	0.00 (0)	0.22

Table 14: Comparison between the four lane detectors on Clip 6.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	82.65	15.31	2.04	0.00 (0)	0.00 (0)	0.58
ALD 2.0	89.91	9.75	0.34	0.00 (0)	0.00 (0)	1.35
ALD 3.0	92.97	6.69	0.34	0.00 (0)	0.00 (0)	1.71
Aly [3]	91.26	4.29	4.45	0.00 (0)	0.00 (0)	0.00

Table 15: Comparison between the four lane detectors on Clip 7.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	96.74	1.68	1.59	0.00 (0)	0.00 (0)	0.80
ALD 2.0	99.56	0.44	0.00	0.00 (0)	0.00 (0)	0.67
ALD 3.0	99.74	0.26	0.00	0.00 (0)	0.00 (0)	0.69
Aly [3]	92.15	6.26	1.59	0.00 (0)	0.00 (0)	1.11

Table 16: Comparison between the four lane detectors on Clip 8.

Algorithm	CO %	MA %	MD %	FD %	TR %	$\sigma_{E(n)}$
ALD 1.0	71.00	27.59	1.42	100.00 (65)	0.00 (0)	0.49
ALD 2.0	91.07	8.93	0.00	100.00 (65)	0.00 (0)	0.81
ALD 3.0	89.99	10.01	0.00	100.00 (65)	0.00 (0)	0.92
Aly [3]	94.02	5.98	0.00	84.51 (54)	15.49 (11)	1.18

From Table 9 - 16, it is clear that all four lane detectors produce good rates for correct detection on most clips. Although the performances of the ALD 2.0, ALD 3.0, and Aly's algorithm [3] are comparable, the ALD 3.0 produces the most number correction detections. Hence, in the rest of this dissertation, future applications of a lane detector will make use of the ALD 3.0.

2.8 Conclusion

This chapter is focused on covering various details of lane detection. Following the introduction, we specified the requirements for a functioning lane detector. Next, we discussed the mechanics behind a few lane detectors that meet the specified requirements in the related work section. Then, we explained the methodologies of three new lane detectors that were introduced in this chapter. Starting with the Advanced Lane Detector (ALD) 1.0, the methodology begins by preprocessing the captured images to highlight the lane markers. Then, using Inverse Perspective Mapping (IPM), the images are geometrically transformed giving them the appearance of a bird's-eye view [20]. Next, an adaptive threshold and a series of template matching procedures are invoked to detect lane markers in the image. Then, using RANSAC (Random Sample Consensus) [23], the outliers among the lane markers are eliminated and the inliers are fitted with a curve that is determined using Linear Least Squares. Finally, the parameters of the curve are tracked from one image to the next using a Kalman filter. The ALD 1.0 worked well when tested with footage containing night time driving, but, it had some problems detecting lane markers during the day. This problem

stems from the adaptive threshold that has difficulty in differentiating between the lane markers from other deformities on the road surface that are visible during the day. This problem limits the application of the ALD 1.0 as a lane detector needs to be able to function in almost any illumination condition. This problem of the ALD 1.0 was the motivation for the development of the ALD 2.0, a new lane detection methodology that was restricted to operating in any particular illumination conditions.

The methodology of the ALD 2.0 begins with IPM where the camera image is converted to a bird's-eye view followed by a color conversion from RGB to YCbCr [38]. Next, Template matching and Hysteresis threshold are used to create binary images in which the lane markers are detected. Then, with the help of the Lane Region Merging block, the binary images that are output from the Filtering block are combined into a single binary image in which lane markers are detected. Then, by subjecting the detected lane markers to certain angular and window constraints, only the lane markers are preserved and the artifacts are suppressed. Finally, using only the detected lane markers, the Piecewise Tracker estimates position of the lane boundary from one image to the next. The ALD 2.0 is an improvement over the ALD 1.0 in that it is able to estimate the lane boundary regardless of the current illumination conditions. However, it is slow, and runs at speeds less than 1 fps. Also, as with the ALD 1.0, it detects the left and the right lane boundaries independently; therefore, it was prone to occasional misalignments that were caused by lens flare, sidewalks and neighboring vehicles. Hence, to help reduce these misalignments, we designed a new lane detection methodology called the ALD 3.0 that detects lane boundaries as a pair of lines that are subject to certain distance and angle constraints.

The methodology of the ALD 3.0 starts by converting the captured images from RGB to grayscale. Next, using IPM, the grayscale image is converted to a bird's-eye. Then, the transformed image undergoes a three stage filtering process that

consists of Template matching, Masking, and Thresholding. The result is a binary image in which the lane markers appear as thin line segments. Then, using the Polar Randomized Hough Transform (PRHT), lines are efficiently detected in the binary image [41]. Once the lines are detected, they are paired based on certain criteria of distance and angular difference between each other. Finally, using a Kalman filter based tracker, the paired lines are tracked from one image to the next. The ALD 3.0 performs well on city roads and highways and in various illumination conditions. It is considered to be an improvement over the ALD 2.0 because not only is it much faster, but it also uses fewer resources to produce nearly identical results. However, it too had minor difficulties with misalignments that could be addressed by appending a few more feature extraction stages to its layout.

Then, for comparison, we also implemented the lane detection algorithm devised by Aly [3] and tested it on the same data that was used to test the ALDs. All four lane detectors performed well but the ALD 3.0 produced the most number of correct detections for the eight clips. Therefore, any future chapters of this dissertation that make use of a lane detector will implement the ALD 3.0.

CHAPTER III

MULTI-CAMERA LANE DEPARTURE WARNING

3.1 Overview

In the previous chapter, we discussed lane detection in detail. We also mentioned that lane detection is a common component in many camera based Driver Assistance (DA) systems. Therefore, we will now take a look at one of the DA systems that relies heavily on lane detection, a Lane Departure Warning system. A Lane Departure Warning (LDW) system informs the driver about a lane change in the near future. Other names for LDW systems used in industry include Lane Departure Prevention (LDP), and Lane Keeping Assist (LKA) systems.

As with a lane detector, the hardware for an LDW system also consists of a forward facing camera that looks out of the front windshield. The images captured by the camera are processed to determine the position of the vehicle in the lane. If the system computes that the vehicle is within a certain distance of the lane boundary, or that the vehicle is going to change lanes in a certain amount of time, then the driver is notified. Upon notification, the driver can take some corrective actions if needed. An illustration of an LDW system is shown in Fig. 137. However, to prevent nuisance from constant signaling, the driver is notified only if certain criteria are not met. A common criterion is the use of the turn signal, which prevents the driver from being notified [46].

An LDW system is often available as a safety feature in select car models from luxury automobile manufacturers. The purpose of an LDW system is to prevent unintentional lane departure. Unintentional departures can be caused by driver distraction, fatigue, or driving under the influence of a controlled substance and can

Figure 137: Illustration of an LDW system that produces an audible warning if the vehicle is about to change lanes [4].

lead to fatal accidents. As a result, to promote safety, OEMs (Original Equipment Manufacturers) such as Mobil-eye have been producing aftermarket products that can be installed in virtually any vehicle [13].

However, existing LDW systems have a serious limitation because of the use of a single camera. To elaborate, while driving during the hours of sunrise and sunset, rays from the sun are likely to be incident directly on the camera lens. As a result, the CCD panel in the camera may saturate and cause the camera to produce images that are unsuitable for use by lane detectors and LDW systems. One technique to eliminate this limitation is by complementing the camera with a second camera. Therefore, in this chapter, we are presenting a new Multi-Camera Lane Departure Warning (MCLDW) system. The MCLDW system uses two cameras with non-overlapping fields of view and facing in opposite directions. One camera looks out of the front windshield and is referred to as the "front camera", and a second camera looks out of the rear windshield and is referred to as the "rear camera". The MCLDW system has two advantages over the single camera LDW system:

1. The images captured by the two cameras can be used to accurately determine the position of the vehicle on the road.

2. If LDW cannot be performed on the images captured by the front camera, images captured by the rear camera are used to perform LDW and vice-versa.

In reference to #2 above, if the front camera is saturated by rays from the sun, LDW can still be performed using images that are provided by the rear camera. In addition, the MCLDW system is able to produce a better estimate of the position of the vehicle in the lane. This results in a more accurate LDW system as will be shown later in this chapter.

Now that we have a brief idea about LDW, in the next section we will define the requirements of an LDW system. Then, we will discuss some of the related research that has been performed in this field. Next, we will introduce the MCLDW system and show how it is used to estimate the position of the vehicle in the lane. Then, if the vehicle is deemed to be under a certain distance from the lane boundary, the driver is informed of a lane departure. Finally, the MCLDW system will be tested with real world data and its performance will be compared to a single camera LDW system.

3.2 Requirements for a Lane Departure Warning System

In this section, we will specify the requirements for a functioning LDW system. The requirements are:

1. It must operate using the lane boundary information provided by a lane detector.

2. Distance between the vehicle and the lane boundaries must be measured, and represented in SI (International System of Units) or Imperial units.

3. Warning should be produced only if the distance is under a threshold. (No turn signal data available).

4. The LDW system is required to operate only on highways at speeds above 40 mph as in many commercial LDW systems [46].

3.3 Related Work

The MCLDW methodology consists of two components:

1. Calibration of non-overlapping camera networks

2. LDW system

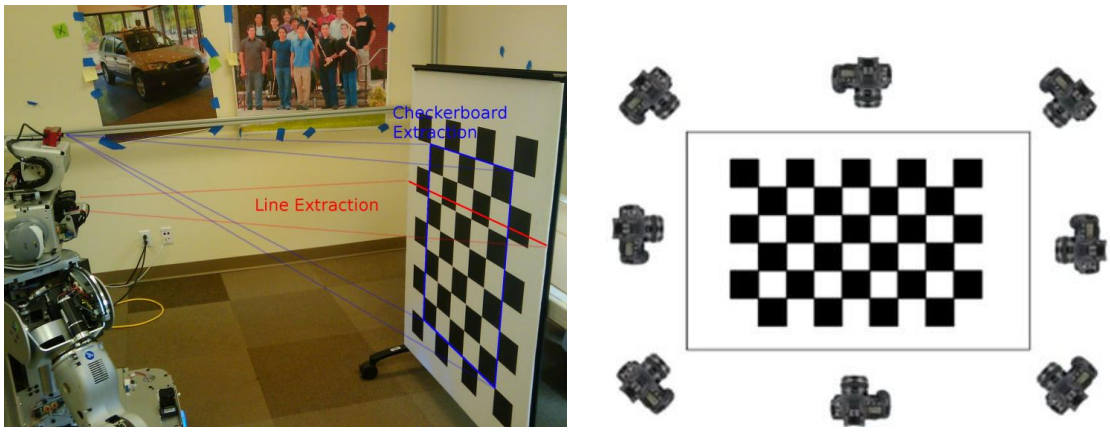
Although extensive research has been performed on both calibration and LDW systems, research on multi-camera LDW systems does not appear to be performed. Therefore, we will take a look at each component independently in the sections below.

3.3.1 Calibration of non-overlapping camera networks

Calibration is defined as the process of estimating the true intrinsic and extrinsic parameters of a camera. The intrinsic parameters of a camera are the field of view, image format, and pixel size. The extrinsic parameters are associated with the camera's position and orientation with respect to an origin and its axes. More specifically, these parameters include the camera's displacement from the origin, yaw, pitch, and roll. However, the intrinsic parameters are already known since they are used to create the world image in Sec. 2.4.2. Similarly, height from the ground, yaw, and pitch of the camera are also known and used in Sec. 2.4.2, while roll is assumed to be zero. Hence, the only unknown parameter is the camera's displacement with respect to the origin. Therefore, in this chapter, **Calibration** is redefined as the process of estimating the physical position of the camera with respect to an origin in the world co-ordinate system.

In the literature, the common approach for calibration is to use a checkerboard

pattern whose position is known and to consider it as a reference. Then, the position of the camera is estimated by determining the transformation that results in the deformed appearance of the checkerboard in the image. If we have a network of cameras; then, this process is repeated for each camera. As a result, the position of each camera can be estimated. An example of the calibration process for a single camera and a camera network is shown in Fig. 138. However, the underlying assumption for



(a) A single camera sees the checkerboard [47].

(b) A network of cameras see the checkerboard [48].

Figure 138: Camera calibration process.

these techniques to work properly is that the cameras must have overlapping fields of view. Basically, the same checkerboard needs to be seen by all the cameras. In the MCLDW system, the two cameras are facing in opposite directions. As a result, the fields of view of the cameras are not overlapping. Therefore, it is impossible for both cameras to see the same checkerboard without making some modifications to the calibration process. However, several novel techniques to calibrate non-overlapping camera networks have been introduced over the years.

In the research shown in [49, 50, 51, 52], cameras are calibrated by tracking targets over multiple camera views. Given the starting point and the known dynamics of the target, calibration is performed by estimating the transformation that led to the appearance of the target in a particular camera's view. However, this research

is performed using cameras that are fixed to a pillar or a building; therefore, their physical positions do not change. In the MCLDW system, the cameras are placed on a mobile platform such as a vehicle; as a result, their positions are constantly changing. Nonetheless, research has also been performed on the calibration of cameras that are mounted on mobile platforms. For example, Pagel [53] has suggested the placement of several unique calibration patterns around the camera network. By identifying the positions of patterns, the network is easily calibrated. However, the author assumes that at least two cameras must be able to see the same pattern. Not only is this not possible in the MCLDW system, but Pagel's [53] camera network is not truly non-overlapping.

Another novel technique is through the use of planar mirrors. In the work shown by Kumar et al. [54] and Lébraly et al. [55], mirrors are placed in a manner such that the reflections of the calibration pattern are visible to the cameras in the network. As a result, by accounting for the positions of the mirrors, the cameras can be calibrated accurately. Although Lébraly et al. hypothesize the application of this technique to calibrate non-overlapping cameras mounted on a vehicle [55], the actual calibration is performed on a compact camera network such as Ladybug2 [56] and in a laboratory environment as also shown by Kumar et al. [54]. This technique is not directly scalable to large size applications such as the one discussed in this dissertation.

Lastly, in the work performed by Lamprecht et al. [5], the two cameras are calibrated by modeling the dynamics of a stationary object on the road. An illustration of this approach is shown in Fig. 139. This approach is similar to the one we will be presenting in the next section. However, in [5], the calibration is performed using only synthetic data without any real world validation. Additionally, Lamprecht et al. also claim that this technique is unable to accurately determine the translation between the two cameras. This is a limitation since the translation between the two cameras needs to be accurately determined. A tabulated comparison between the

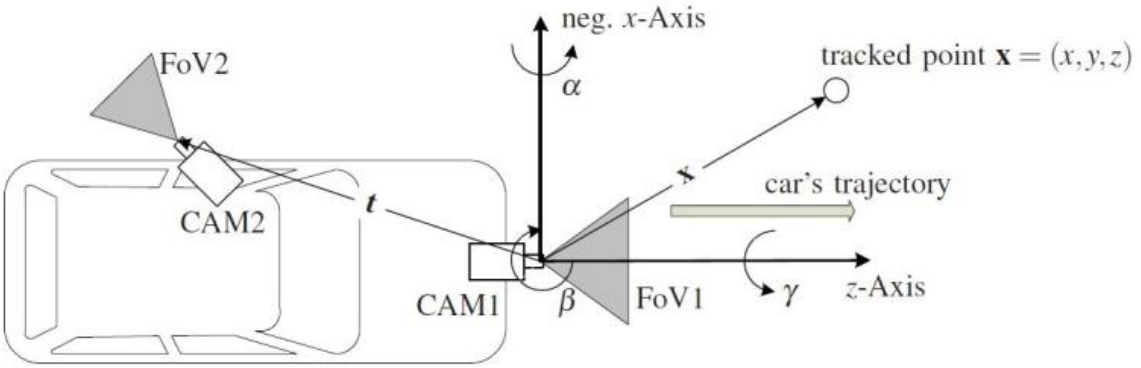


Figure 139: Calibration of non overlapping cameras by tracking a stationary object on the road [5]. The object is shown near the top right and labeled **x**. The two cameras are labeled CAM1 and CAM2.

different techniques used to perform calibration of non-overlapping camera networks is shown in Table 17.

Table 17: Comparison between the different techniques used to calibrate camera networks.

Authors	Camera Installation			Test / Validation Data	
	Fixed	Mobile Platform			
		Lab	Vehicle	Synthetic	Real-world
Rahimi et al. [49]	Y	N	N	Y	Y
Shafique et al. [50]	Y	N	N	N	Y
Tiue et al. [51]	Y	N	N	N	Y
Wang et al. [52]	Y	N	N	Y	N
Pagel [53]	N	Y	N	Y	Y
Kumar et al. [54]	N	Y	N	N	Y
Lébraly et al. [55]	N	Y	N	N	Y
Lamprecht et al. [5]	N	N	N	Y	N
Borkar et al. [57]	N	N	Y	N	Y

3.3.2 LDW system

Over the years, several techniques have been introduced to perform LDW. However, the four techniques that have appeared often in literature are:

1. Angle of Lane Boundary (ALB)

2. Changing Rate of Angles between Lanes (CRAL)

3. Time to Lane Crossing (TLC)

4. Current Car Position (CCP)

The ALB is a very simple approach to LDW. Using a straight line to model the lane boundary, the angle of the estimated lane boundary is monitored. If this angle is within a range of values, then a warning is produced to inform the driver of an imminent lane change [58, 59, 60, 61]. Unfortunately, in ALB, the angle of the lane boundaries is measured and not the physical distance to the lane boundaries. As a result, it does not meet our all requirements in Sec. 3.2; hence, it is not considered in the rest of this dissertation.

CRAL is an extension of the ALB. In CRAL, the angle of intersection of the two lane boundaries at the horizon is computed. Then, this computed angle is used as an estimate of the position of the vehicle on the road. An LDW signal is produced if the angle is within a predefined range of values [62, 63, 64]. As with the ALB, CRAL does not meet our all requirements in Sec. 3.2 since the angle of the lane boundaries is computed and not the physical distance to the lane boundaries. Hence, it is not considered in the rest of this dissertation.

TLC is another popular technique that is used in LDW systems. TLC operates by computing the time before the vehicle crosses either the left or the right lane boundaries. Using camera calibration and dynamics of the vehicle, this time is computed. Furthermore, if the computed time is below a certain threshold, then the driver is informed that a lane change is imminent [65, 66, 67, 68]. TLC is expected to operate well on straight roads but may face difficulty on curving roads. This is because many existing LDW systems make use of lane detectors that model the lane boundaries as straight lines. As a result, estimating the lane boundaries on curving roads may be problematic as discussed in Sec. 2.6.7. Consequently, the computed time to cross the lane boundary may be inaccurate. Additionally, TLC does not compute the immediate distance between the vehicle and the lane boundary. Therefore, as with CRAL

and ALB, TLC also does not all meet our requirements from Sec. 3.2, hence, it is not considered from here on. This brings us to the CCP.

In CCP, the distance between the vehicle and lane boundary is determined as shown in Fig. 140. Therefore, if the distance is estimated to be under a threshold, the driver is notified of an imminent lane change [69, 70, 71, 6]. Unlike CRAL and

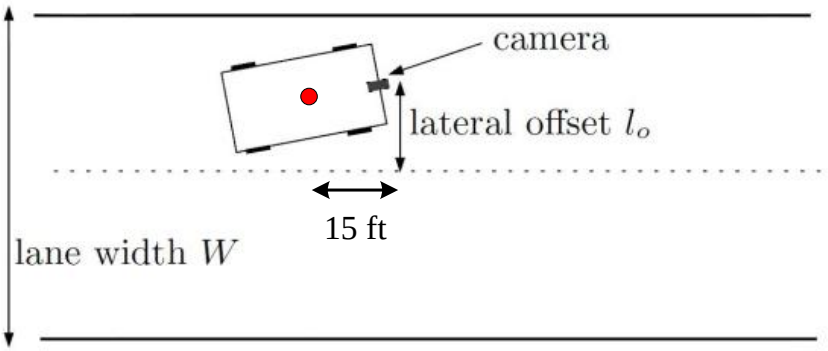


Figure 140: CCP measures the distance to the lane boundary [6]. The red dot in the image is the center of the vehicle.

TLC, CCP meets all our requirements stated in Sec. 3.2. However, CCP is often implemented using a single camera configuration; therefore, the measured distance is not the immediate distance between the vehicle and lane boundary, rather it is about 15 ft ahead of the vehicle's center. This is due to camera geometry and the visibility of the road from the position of the camera as shown in Fig. 141. Ideally, with extrapolation, the immediate distance or distance alongside the vehicle can be determined. However, as most LDW system use straight line models to estimate the lane boundaries, the extrapolated location of the lane boundary may not be very accurate. This is a limitation since the accurate estimation of the distance between the vehicle and the lane boundary is important. A tabulated comparison between the different techniques is shown in Table 18.

Based on the discussion above, it is clear that extensive research has been performed on both calibration and LDW systems. However, it does not appear that has

Figure 141: Distance from the front of the vehicle to different locations on the road. The lowest row in the image that corresponds to the road surface is about 10 ft ahead of the vehicle's front bumper.

Table 18: Comparison between the different techniques used for Lane Departure Warning (LDW).
[1]A:Angle of lane boundary, T:Time before lane is crossed, D:Distance to lane boundary.

Technique	Used by	Camera calibration needed?	Warning based on[1]	Location of measured distance
ALB	[58, 59, 60, 61]	N	A	N/A
CRAL	[62, 63, 64]	N	A	N/A
TLC	[65, 66, 67, 68]	Y	T	N/A
CCP	[69, 70, 71, 6]	Y	D	$\sim$ 10 ft ahead
MCLDW	[57]	Y	D	Immediate

research been performed on multi-camera LDW systems. This is the novelty of the MCLDW system [57]. The MCLDW system improves the existing CCP approach by adding a second camera to look out of the rear windshield. Then, lane detection is also performed on the images captured by the rear camera. Thus, we have the estimates of the lane boundaries in the images of both the front and the rear camera. Using the calibration details of each camera, and the estimates of the lane boundaries, a smooth spline that connects the lane boundaries from the front to the rear view

is computed. Furthermore, evaluating the spline at a specific point in the world co-ordinates allows us to determine the immediate distance between the vehicle and the lane boundary. In contrast to CCP, this is a more accurate estimate of the vehicle's position in the lane. The idea of connecting the lane boundaries between the two cameras is similarly shown in the research performed by Ieng et al. [72]. However, in [72], the authors provide no technical details on how the cameras are calibrated, or how the lane boundaries are connected. The information is terse and simple images are used to convey the message. As a result, we found this work to be incomplete.

3.4 Camera Calibration

In this section, we will explain the procedure to determine the physical position of each camera that is placed in the vehicle. The position is denoted using a position vector which is described as

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{69}$$

where x, y, and z are the x, y, and z co-ordinates in the world, respectively. Since calibration involves determining the position of the camera with respect to an origin in the world co-ordinate system, an origin needs to be specified. The origin is chosen as the center of the vehicle as shown in Fig. 142 and denoted as $\mathbf{P_o}$ where

$$\mathbf{P_o} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{70}$$

Also shown in Fig. 142 are $\mathbf{P_f}$ and $\mathbf{P_r}$, the positions of the front and the rear cameras, respectively. Furthermore, the front and the rear cameras face in opposite directions. For calibration, we used images that were captured simultaneously by both the front and the rear camera at 30 fps. Images captured by the front camera are denoted as $I_{f,n}$, and images captured by the rear camera are denoted as $I_{r,n}$. In the notation,

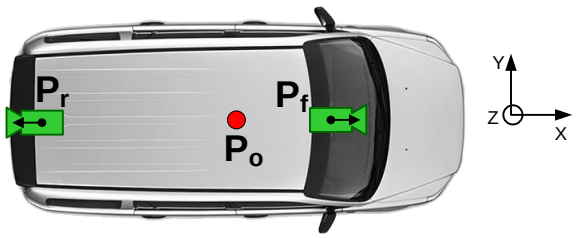


Figure 142: Position of the origin and two cameras.

n is the image index. Furthermore, these images were captured while driving at a constant speed and on a straight road where solid lane boundaries are visible. In these images, we were able to model the dynamics of the solid lane markers on the road. Then, using the dynamics of the lane markers, we were able to determine the displacement between the cameras. Sample images captured by the two cameras that are used in calibration are shown in Fig. 143. Along with the images, we also record the speed of the vehicle.

(a) Front camera image $I_{f,n}$.

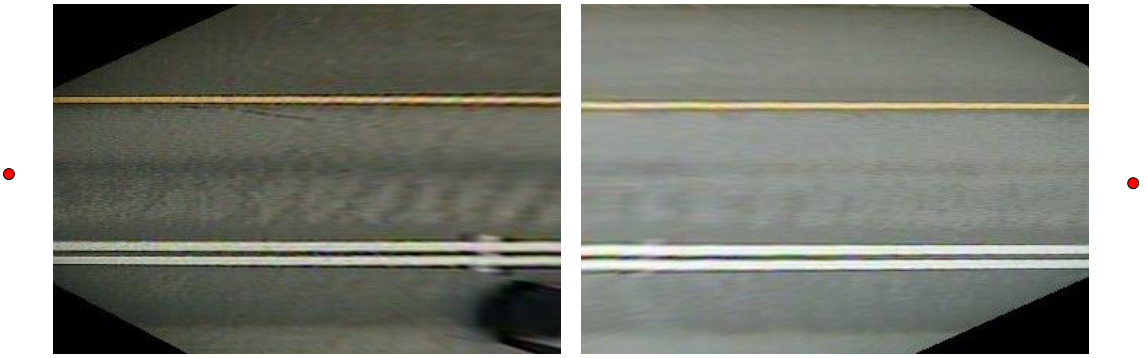
(b) Rear camera image $I_{r,n}$.

Figure 143: Sample images used in calibration.

Since the height of the camera from the ground ($z = 0$) is measured along the z axis, the camera's z co-ordinate is the same as its height from the ground. Therefore, based on the discussion in Sec. 3.3.1, the z co-ordinates of $\mathbf{P_f}$ and $\mathbf{P_r}$ are known. Hence, the techniques described below are used to determine only the x and the y co-ordinates of $\mathbf{P_f}$ and $\mathbf{P_r}$.

3.4.1 Offset between the cameras

From the discussions in the previous sections, we know the intrinsic and a few of the extrinsic parameters of both cameras. Therefore, with these parameters, we create world images. World images are created by applying the IPM transformation (details in Sec. 2.4.2) to the images captured by each camera. Moreover, the world image created from $I_{f,n}$ is referred to as $W_{f,n}$. Similarly, $W'_{r,n}$ is the world image created from $I_{r,n}$. Since the rear camera faces in the opposite direction, $W'_{r,n}$ is rotated 180° about its origin to create $W_{r,n}$. In Fig. 144, we have shown images of $W_{f,n}$ and $W_{r,n}$. In $W_{f,n}$, the origin is to the left; therefore, each image pixel in $W_{f,n}$ has positive x



(a) $W_{f,n}$ created by transforming $I_{f,n}$. The red dot on the left is the optical center.

(b) $W_{r,n}$ created by transforming $I_{r,n}$ and then rotating the image by 180°. The red dot on the right is the optical center.

Figure 144: World images created from the camera images.

co-ordinates. On the other hand, in $W_{r,n}$, the origin is to the right. Therefore, each image pixel in W_r has negative x co-ordinates. Next, $W_{f,n}$ and $W_{r,n}$ are stacked side by side as shown in Fig. 145. If the positions of the two cameras do not have the same y co-ordinate, a visible offset is observed between the lane markers. This is illustrated in Fig. 145. We assume that the y co-ordinate of $\mathbf{P_f}$ is aligned with $\mathbf{P_o}$ i.e. $y = 0$ in $\mathbf{P_f}$. Therefore, the offset is adjusted by translating the $W_{r,n}$ in the direction of the y axis until the lane markers line up correctly. Once $\mathbf{P_f}$ and $\mathbf{P_r}$ are aligned correctly, the lane markers appear as long lines that seamlessly connect from $W_{f,n}$ to $W_{r,n}$ as shown in Fig. 146.

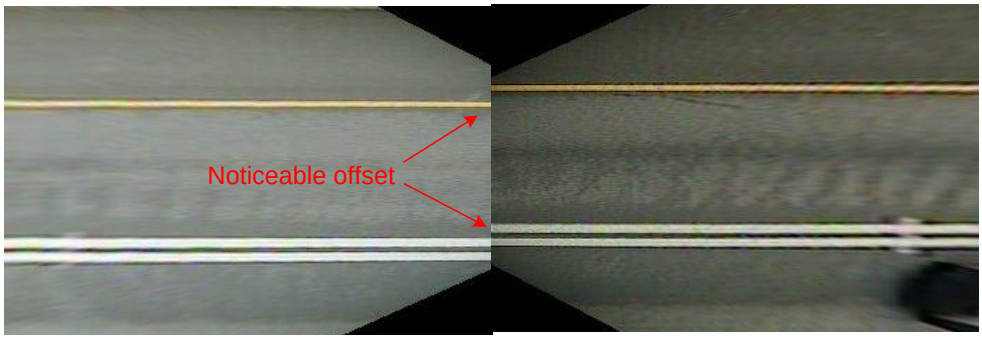


Figure 145: An offset is visible when $W_{f,n}$ and $W_{r,n}$ are stacked side by side.

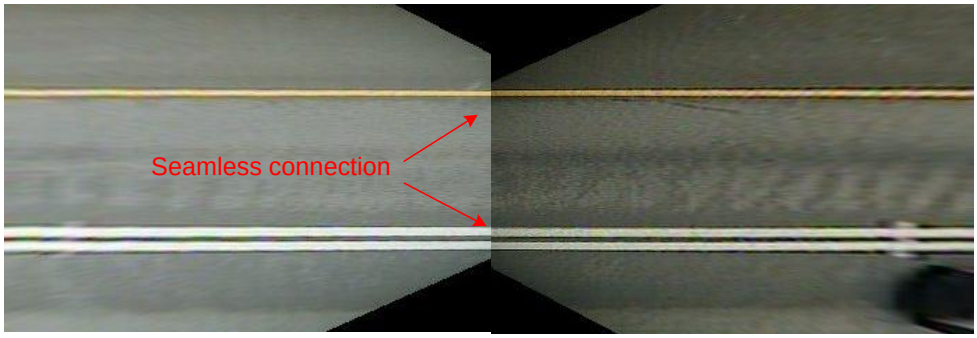


Figure 146: After $W_{f,n}$ and $W_{r,n}$ are aligned correctly, the lane markers appear as long lines.

3.4.2 Distance between the cameras

The distance between the two cameras is estimated by tracking a reference marking on the road surface that first appears in ${W_f}^1$, and then appears in ${W_r}^1$ after a number of images. In our case, we use the diamond from the High Occupancy Vehicle (HOV) lane as the reference marking. We selected this particular marking because it has four corners, this makes it easy to track several points on the marking. An HOV diamond with annotated corners is shown in Fig. 147. If HOV diamonds are not present in the calibration images, then we can use certain types of lane markers as the reference markings. However, we cannot use solid lane markers because they do not have corners. Hence, it is not possible to track a specific point on the lane marker

[1] W_f is used to denote a world image created from an image captured by the front camera. This notation is used purely for explanation does not correspond to a particular image index; hence, n is dropped from the notation. Similarly, W_r applies to a world image created from an image captured by the rear camera.

as shown in Fig. 147. Fortunately, a dashed lane marker has corners and can be used as the reference marking as shown in Fig. 147. However, one must be careful

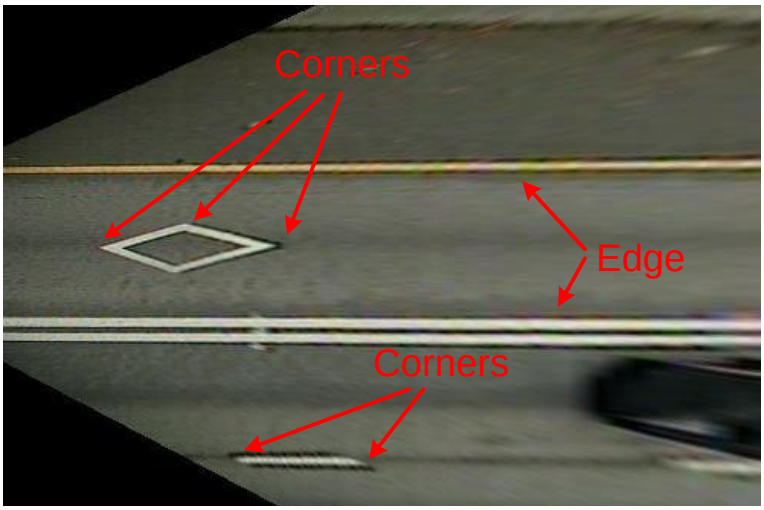


Figure 147: HOV diamond, solid lane marker and dashed lane marker with corners and edges annotated.

when using dashed lane markers. The reason for this is that dashed lane markers occur every $30-40$ ft on the road [19]; hence, they appear very frequently in W_f and W_r. Therefore, it can become quite difficult to track a particular lane marker that has appeared in W_f and then reappears in W_r. In contrast, the HOV diamonds are spaced at least 1000 ft apart [19]. Hence, they appear far less frequently in W_f or W_r in comparison to the lane markers. As a result, tracking a particular HOV diamond that first appears in W_f and then reappears in W_r is trivial. This makes it possible to track a specific corner on the diamond.

First, we observe the diamond that appears in $W_{f,n}$ and estimate the x co-ordinate of the right most corner on the diamond. From this co-ordinate, it is possible to compute the horizontal distance k, the distance between the corner of the HOV diamond and the camera's optical center as shown in Fig. 148. The optical center is the location of the origin in W_f. Since the lens used by this camera has a focal length of 5mm, it is safe to assume that the optical center is the same as the camera's position i.e. $\mathbf{P_f}$. We only compute the horizontal distance between the two locations because

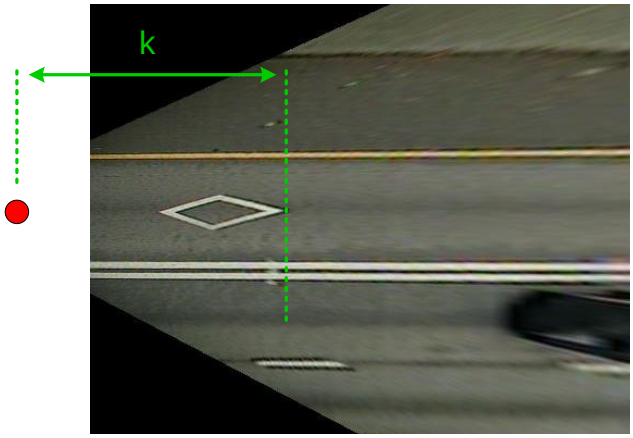


Figure 148: Determining the distance between the HOV diamond and the front camera $\mathbf{P_f}$ in $W_{f,n}$.

it is assumed that the vehicle is travelling on a straight road (details in Sec. 3.4).

Similarly, we observe the diamond that has reappeared in $W_{r,n+\Delta}$, which is a delayed version of $W_{r,n}$ that is created from $I_{r,n+\Delta}$. We observe the diamond in $W_{r,n+\Delta}$ instead of $W_{r,n}$ because $W_{f,n}$ and $W_{r,n}$ are created from $I_{f,n}$ and $I_{r,n}$, respectively. These images are captured simultaneously; therefore, the same diamond from $W_{f,n}$ will not appear in $W_{r,n}$. However, it will appear in W_r after Δ images during which the vehicle will have moved forward. Again, we estimate only the x co-ordinate of the right most corner of the diamond. From this co-ordinate, we are able to compute the horizontal distance j, the distance between the corner of the HOV diamond and the camera's optical center as shown in Fig. 149. Since we use the same lens that is used by $\mathbf{P_f}$, it is also safe to assume that the optical center is the same as the camera's position i.e. $\mathbf{P_r}$.

We now know the distances between the right most corner of the diamond and the cameras in $W_{f,n}$ and $W_{r,n+\Delta}$. $W_{f,n}$ and $W_{r,n+\Delta}$ are placed side by side with a gap in between them as shown in Fig. 150. The two red dots in Fig. 150 are the positions of the two cameras, $\mathbf{P_f}$ and $\mathbf{P_r}$. Since Δ is known, we can compute the distance travelled by the vehicle in Δ images. This distance is equivalent to w, the distance between the two diamonds shown in Fig. 150. Since we also know the frame rate of

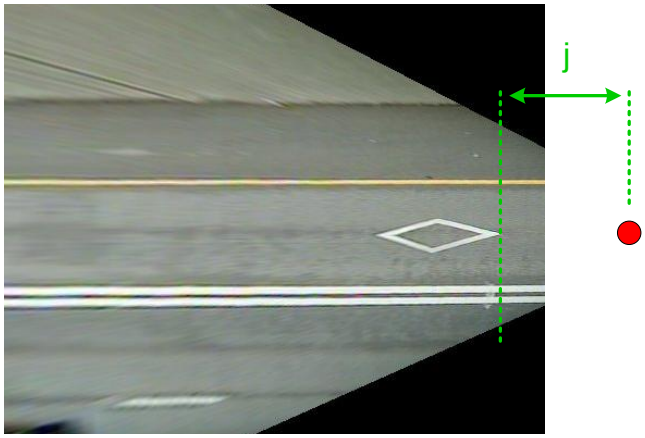


Figure 149: Determining the distance between the HOV diamond and the rear camera $\mathbf{P_r}$ in $W_{r,n+\Delta}$.

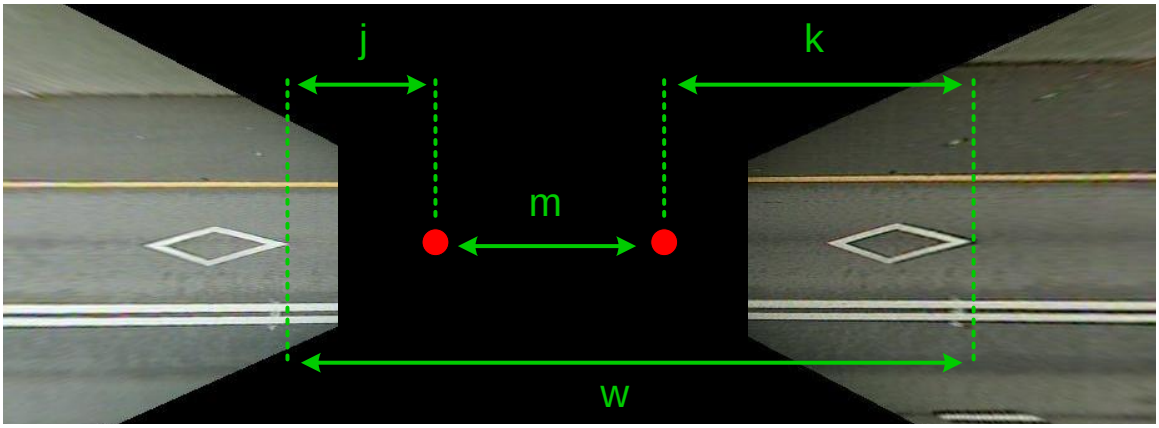


Figure 150: Determining the distance between the two cameras.

the images, and the speed of the vehicle, w is computed as

$$w = \text{speed} \times \frac{\Delta}{30 \text{ fps}} \tag{71}$$

However, Speed is recorded in units of mph (miles per hour). To convert speed into units of feet per second, Eq. (71) is multiplied by a constant as shown below

$$w = \text{speed} \times \frac{5280}{3600} \times \frac{\Delta}{30 \text{ fps}} \tag{72}$$

To compute the distance between the two cameras, we compute m as

$$m = w - (j + k) \tag{73}$$

However, m is the distance between the two cameras. The value of m does not specify the actual positions of the cameras on the x axis. Therefore, we measured $x_f = 2$, the

x co-ordinate of $\mathbf{P_f}$ with respect to $\mathbf{P_o}$. Consequently, $x_r = x_f - m$, the x co-ordinate of $\mathbf{P_r}$ with respect to $\mathbf{P_o}$. Upon completing the calibration steps, we determined

$$\mathbf{P_f} = \begin{bmatrix} 2 \\ 0 \\ 4.81 \end{bmatrix}, \mathbf{P_r} = \begin{bmatrix} -7.5 \\ -0.49 \\ 4.91 \end{bmatrix} \tag{74}$$

The units of $\mathbf{P_f}$ and $\mathbf{P_r}$ are ft.

3.5 Multi-Camera Lane Departure Warning System

In this section, we introduce the Multi-Camera Lane Departure Warning (MCLDW) system. Unlike existing LDW systems, the MCLDW system uses two cameras facing in opposite directions. The data from the two cameras is combined to accurately estimate the vehicle's position in the lane. Based on this estimate, the driver is informed of a lane change. The layout of the MCLDW system is shown in Fig. 151. First, lane detection is performed on the images captured by both the front and the rear cameras. Then, the detected lane boundaries undergo a co-ordinate system adjustment based on the calibration details of each camera. Upon completing the adjustment, the

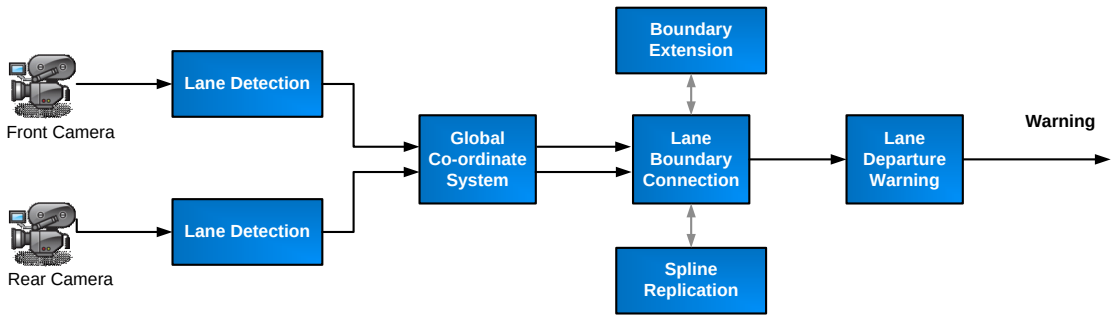


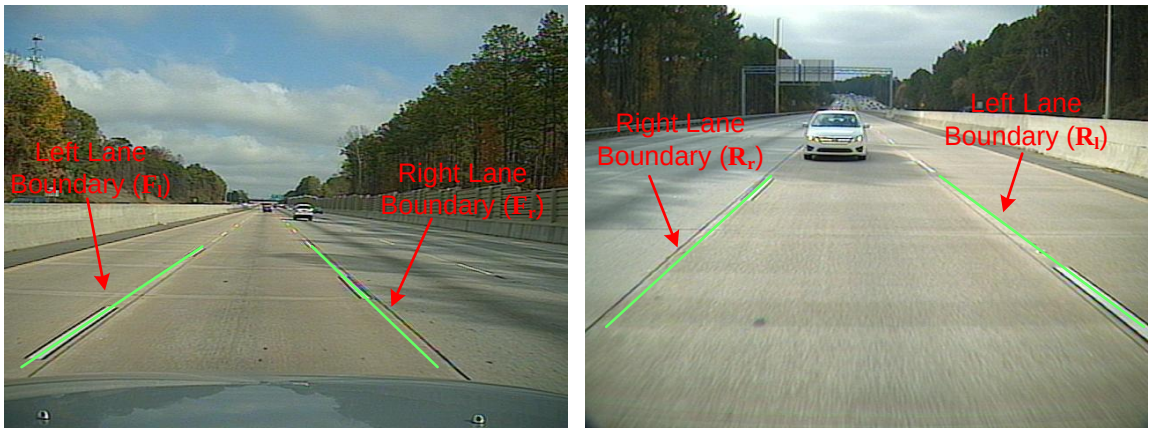
Figure 151: Layout of the MCLDW system.

lane boundary data from both cameras is combined to produce estimates of the lane boundaries alongside the vehicle. In the case that one of the two cameras falls short in performance and cannot detect the lane boundaries, either the spline replication or the boundary extension process as described in Sec. 3.5.4 are used to estimate the

lane boundaries. Using these estimates, the distance between the vehicle and the lane boundaries is determined. If this distance on either side of the vehicle is under a threshold, then the driver is informed of a lane change. The details of each block in the layout of the MCLDW system are provided below.

3.5.1 Lane Detection

Lane detection involves the estimation of the lane boundaries in an image. Since lane detection has been explained previously in this dissertation, please refer to Chap. 2 for complete details. Using the Advanced Lane Detector 3.0 (details in Sec. 2.6), lane boundaries in both $I_{f,n}$ and $I_{r,n}$ are estimated as shown in Fig. 152. The estimates of



(a) Left and right lane boundaries detected in $I_{f,n}$.

(b) Left and right lane boundaries detected in $I_{r,n}$. Notice that the left and right lane boundaries are on opposite sides.

Figure 152: Lane boundaries estimated in the images of the front and the rear cameras.

the lane boundaries are provided as a list of points in matrix notation

$$\mathbf{F_l} = \begin{bmatrix} r_1 & c_1 \\ r_2 & c_2 \\ \vdots & \vdots \\ r_m & c_m \end{bmatrix} \tag{75}$$

where $\mathbf{F_l}$ contains the (r, c) co-ordinates of each point on the left lane boundary in $I_{f,n}$. Similarly, $\mathbf{F_r}$ contains the (r, c) co-ordinates of each point on the right lane

boundary in $I_{f,n}$. $\mathbf{R_l}$ and $\mathbf{R_r}$ contain the (r, c) co-ordinates of each point on the left and the right lane boundary, respectively in $I_{r,n}$.

3.5.2 Global Co-ordinate System

Since we are using multiple cameras, it is essential to establish a global co-ordinate system. With a global co-ordinate system, data from the other cameras in the network can be combined meaningfully. Otherwise, the data is only relative to the views of the respective cameras.

As stated in Sec. 3.4, $\mathbf{P_o}$ is selected as the origin as shown in Fig. 142. $\mathbf{P_f}$ and $\mathbf{P_r}$ are the positions of the two cameras in Fig. 142. Since the positions of the cameras are relative to $\mathbf{P_o}$, we can specify a translation $\mathbf{t_f}$ that translates $\mathbf{P_o}$ to $\mathbf{P_f}$. This translation is given as

$$\mathbf{t_f} = \mathbf{P_f} - \mathbf{P_o} \tag{76}$$

Similarly, we can determine a translation $\mathbf{t_r}$ that translates $\mathbf{P_o}$ to $\mathbf{P_r}$ which is given as

$$\mathbf{t_r} = \mathbf{P_r} - \mathbf{P_o} \tag{77}$$

Next, we create a empty world image $W_{g,n}$. We then specify the location of $\mathbf{P_o}$ in image. Using $\mathbf{t_f}$ and $\mathbf{t_r}$, we can determine the locations of $\mathbf{P_f}$ and $\mathbf{P_r}$ in the image relative to $\mathbf{P_o}$ as shown in Fig. 153. Since $\mathbf{P_f}$ and $\mathbf{P_r}$ are the optical centers of $W_{f,n}$

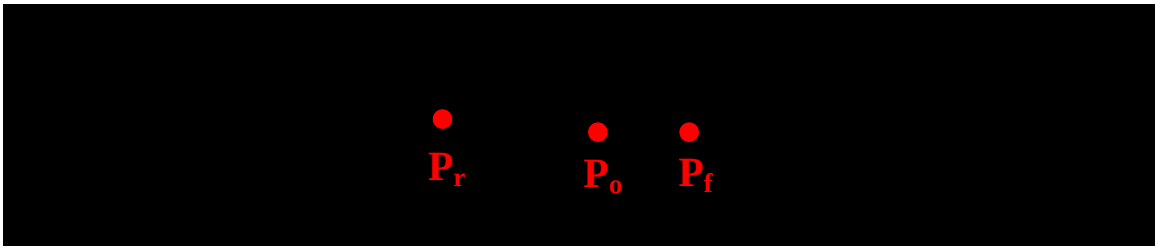


Figure 153: Locations of $\mathbf{P_o}$, $\mathbf{P_f}$, and $\mathbf{P_r}$ in $W_{g,n}$.

and $W_{r,n}$, respectively, we can appropriately place $W_{f,n}$ and $W_{r,n}$ in $W_{g,n}$ as shown in Fig. 154.

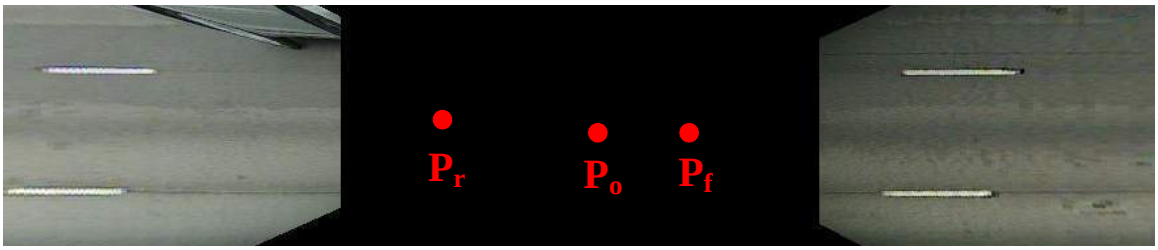


Figure 154: Transformed $W_{f,n}$ appears on the right and the transformed $W_{r,n}$ appears on the left in $W_{g,n}$.

In Sec. 3.5.1, we defined $\mathbf{F_l}$, $\mathbf{F_r}$, $\mathbf{R_l}$, and $\mathbf{R_r}$ as the matrices containing the list of points that represent the estimates of the lane boundaries. However, these points correspond to locations in a camera image and need to be transformed to world co-ordinates. Thus, by plugging the (r, c) co-ordinates of each point in $\mathbf{F_l}$ into Eq. (29)-(31), we create

$$\mathbf{F_{w,l}} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \end{bmatrix} \tag{78}$$

which contains the (r, c) co-ordinates that have been transformed to world co-ordinates (x, y). Similarly, $\mathbf{F_{w,r}}$, $\mathbf{R_{w,l}}$, and $\mathbf{R_{w,r}}$ are created by plugging the (r, c) co-ordinates in $\mathbf{F_r}$, $\mathbf{R_l}$, and $\mathbf{R_r}$, respectively, into Eq. (29)-(31).

$\mathbf{F_{w,l}}$ and $\mathbf{F_{w,r}}$ are created under the assumption that the x and the y co-ordinate of $\mathbf{P_f}$ are at the origin. In $W_{g,n}$, the origin is at $\mathbf{P_o}$. Therefore, $\mathbf{F_{w,l}}$ is translated using $\mathbf{T_f}$. As a result, we get

$$\mathbf{F_{g,l}} = \mathbf{F_{w,l}} + \mathbf{T_f} \tag{79}$$

which contains the estimates of the left lane boundary in $W_{f,n}$, but with respect to the origin $\mathbf{P_o}$ in the global co-ordinate system. In Eq. (79),

$$\mathbf{T_f} = \begin{bmatrix} x_f & y_f \\ x_f & y_f \\ \vdots & \vdots \end{bmatrix} \tag{80}$$

where x_f and y_f are the x and y co-ordinates, respectively, in $\mathbf{P_f}$. $\mathbf{T_f}$ has dimensions $m \times 2$ where m is equal to the number of rows in $\mathbf{F_{w,l}}$. Similarly, $\mathbf{F_{g,r}}$ is created as

$$\mathbf{F_{g,r}} = \mathbf{F_{w,r}} + \mathbf{T_f} \tag{81}$$

The creation of $\mathbf{R_{g,l}}$ and $\mathbf{R_{g,r}}$ is slightly different. In the creation of $W_{r,n}$, $W'_{r,n}$ is rotated by 180°. Hence, $\mathbf{R_{w,l}}$ and $\mathbf{R_{w,r}}$ are also rotated by 180°. This is because the lane detector in Sec. 3.5.1 estimates lane boundaries in $W'_{r,n}$ (by transforming $I_{r,n}$) and not $W_{r,n}$. Therefore, $\mathbf{R_{g,l}}$ is created as

$$\mathbf{R_{g,l}} = \mathbf{A} \cdot \mathbf{R_{w,l}} + \mathbf{T_r} \tag{82}$$

which contains the estimates of the left lane boundary in $W_{r,n}$, also with respect to the origin $\mathbf{P_o}$ in the global co-ordinate system. In Eq. (82), $\mathbf{A}$ is a 180° rotation matrix and

$$\mathbf{T_r} = \begin{bmatrix} x_r & y_r \\ x_r & y_r \\ \vdots & \vdots \end{bmatrix} \tag{83}$$

where x_r and y_r are the x and y co-ordinates, respectively, in $\mathbf{P_r}$. $\mathbf{T_r}$ has dimensions $m \times 2$ where m is equal to the number of rows in $\mathbf{R_{w,l}}$. Similarly, $\mathbf{R_{g,r}}$ is created as

$$\mathbf{R_{g,r}} = \mathbf{A} \cdot \mathbf{R_{w,r}} + \mathbf{T_r} \tag{84}$$

The transformed lane boundaries as they appear in $W_{g,n}$ are shown in Fig. 155.

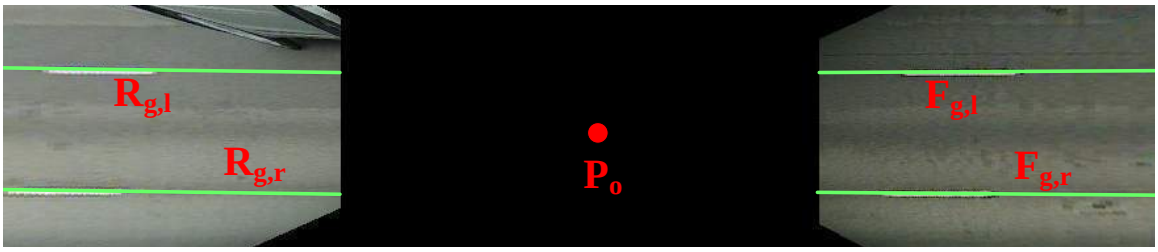


Figure 155: Transformed lane boundaries as they appear in $W_{g,n}$.

3.5.3 Lane Boundary Connection

Now that we have the estimates of the lane boundaries in $W_{g,n}$, we can connect the lane boundaries from $W_{f,n}$ to $W_{r,n}$. The connection is created using a Catmull-Rom spline. A Catmull-Rom spline is a type of cubic Hermite spline that produces a smooth curve within the interval of two control points [73]. We use this spline because it produces a good estimate of the lane boundary that is outside of both cameras fields of view. The Catmull-Rom spline that estimates the lane boundary is shown in Fig. 156. In the image, $\mathbf{Q_{r,1}}$ and $\mathbf{Q_{r,2}}$ are the two end points of the right lane boundary in $W_{f,n}$. Similarly $\mathbf{Q_{r,3}}$ and $\mathbf{Q_{r,4}}$ are the two end points of the right lane boundary in $W_{r,n}$.

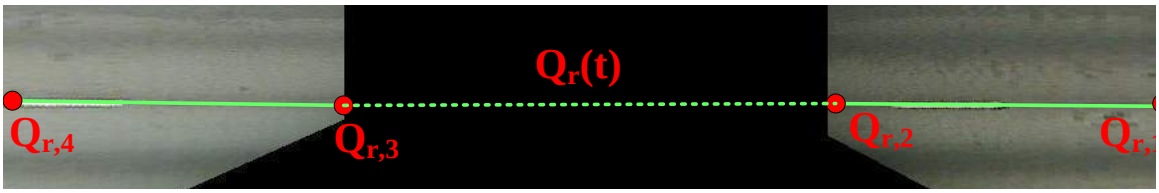


Figure 156: Using the Catmull-Rom spline, the lane boundary that is outside of the cameras fields of view is estimated.

Besides the Hermite spline, other techniques such as linear interpolation and B-splines can also be used to produce the connection between the two lane boundaries. However, a linear interpolation will produce a straight line connection. A straight line is a good estimate of the lane boundary on a straight road, but not a curving road. Similarly, the B-spline is an approximating spline; as a result, the generated curve may not pass through $\mathbf{Q_{r,2}}$ and $\mathbf{Q_{r,3}}$ in the image. We believe this to be an inaccurate estimate of the lane boundary. On the other hand, Catmull-Rom spline is an interpolating spline. As a result, the generated curve will always pass through $\mathbf{Q_{r,2}}$ and $\mathbf{Q_{r,3}}$.

To create the Catmull-Rom spline, four points are needed. These points are shown in Fig. 156 and labeled $\mathbf{Q_{r,1}} - \mathbf{Q_{r,4}}$. Of these four points, $\mathbf{Q_{r,2}}$ and $\mathbf{Q_{r,3}}$ are the

controls points between which the curve is estimated. Then, $\mathbf{Q_{r,1}Q_{r,3}}$ and $\mathbf{Q_{r,2}Q_{r,4}}$ form tangents that are used to "guide" the spline between the two control points. The parametric representation of the spline for the right lane boundary is given as

$$\mathbf{Q_r}^T(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 1 & -\frac{3}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \mathbf{Q_{r,1}}^T \\ \mathbf{Q_{r,2}}^T \\ \mathbf{Q_{r,3}}^T \\ \mathbf{Q_{r,4}}^T \end{bmatrix} \tag{85}$$

where $t \in [0, 1]$ is the normalized unit interval between $\mathbf{Q_{r,2}}$ and $\mathbf{Q_{r,3}}$. In the above equation, $\mathbf{Q_{r,i}}$ is a column vector and written as

$$\mathbf{Q_{r,i}} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{86}$$

where x and y are the x and the y co-ordinates of a point on the lane boundary. Therefore, $\mathbf{Q_{r,1}}$ corresponds to the point in $\mathbf{F_{g,r}}$ that is the farthest away from $\mathbf{P_f}$ along the x axis as shown in Fig. 156. Similarly, $\mathbf{Q_{r,2}}$ corresponds to the point in $\mathbf{F_{g,r}}$ that is nearest to $\mathbf{P_f}$ along the x axis. $\mathbf{Q_{r,3}}$ and $\mathbf{Q_{r,4}}$ correspond to points that are the nearest and farthest from $\mathbf{P_r}$ in $\mathbf{R_{g,r}}$, respectively.

Above, we explained how we created a spline $\mathbf{Q_r}(t)$ to estimate the right lane boundary that is out of both cameras fields of view. To create a spline $\mathbf{Q_l}(t)$ for the left lane boundary, we repeat the above process except, $\mathbf{Q_{r,1}} - \mathbf{Q_{r,4}}$ in Eq. (85) are replaced with $\mathbf{Q_{l,1}} - \mathbf{Q_{l,4}}$. These four points are acquired from $\mathbf{F_{g,l}}$ and $\mathbf{R_{g,l}}$. The spline curve estimating the left lane boundary is shown in Fig. 157.

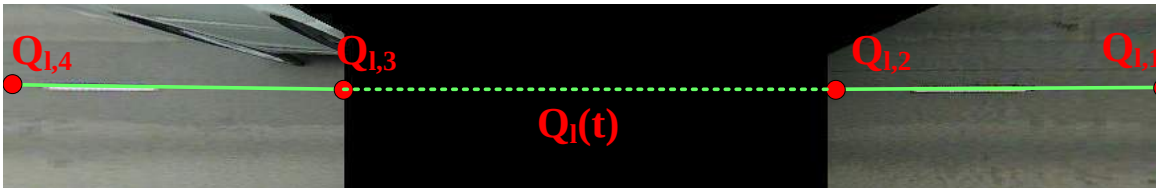


Figure 157: Catmull-Rom spline is used to estimate $\mathbf{Q_l}(t)$.

3.5.4 Spline Replication and Boundary Extension

In the previous section, we showed that a spline is used to estimate the lane boundaries. Splines work well when both the left and the right lane boundaries are detected in $I_{f,n}$ and $I_{r,n}$ as shown in Fig. 152. This is considered as the default mode of operation of the MCLDW system. However, there are some occasions when the performance of one of the two cameras falls short. For example, in Fig. 158, both the left and the right lane boundaries are detected in $I_{f,n}$. But, in $I_{r,n}$ only the right lane boundary is detected. In this situation, a spline cannot be created since only

(a) Left and right lane boundaries are detected in $I_{f,n}$.

(b) Only the right lane boundary is detected in $I_{r,n}$.

Figure 158: Lane boundaries estimated in the images of the front and the rear cameras.

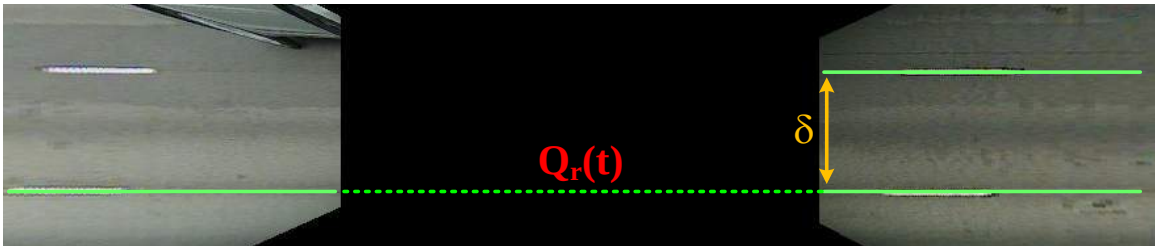
two points are available in the estimation of the left lane boundary. Hence, to handle these situations, we introduce two alternate modes of operation:

1. Spline replication

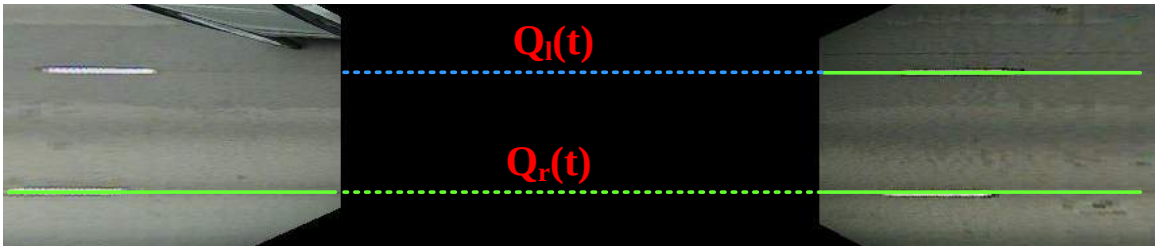
2. Boundary extension

Spline replication is the process of using redundant lane boundary information to complete missing data. To elaborate, let us consider the example above where both the left and the right lane boundaries are detected in $I_{f,n}$, while only the right

lane boundary is detected in $I_{r,n}$. A spline $\mathbf{Q_r}(t)$ is created to estimate the right lane boundary outside of the camera views. However, the absence of the left lane boundary in $I_{r,n}$ prevents the creation of the spline $\mathbf{Q_l}(t)$. In this situation, the estimated right lane boundary is replicated on the left side of the vehicle and offset by the distance δ. This idea is illustrated in Fig. 159a where the spline $\mathbf{Q_r}(t)$ is shown in green. $\mathbf{Q_r}(t)$ is copied and offset by δ to the left side as depicted by the blue spline in Fig. 159b.



(a) Spline $\mathbf{Q_r}(t)$ is created.



(b) $\mathbf{Q_r}(t)$ is copied to the left and offset by δ to create $\mathbf{Q_l}(t)$.

Figure 159: Spline replication process.

δ is computed as

$$\delta = \begin{bmatrix} \mathbf{Q_{l,2}}^T - \mathbf{Q_{r,2}}^T \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{87}$$

where $\mathbf{Q_{l,2}}$ and $\mathbf{Q_{r,2}}$ are the two points in $\mathbf{F_{g,l}}$ and $\mathbf{F_{g,r}}$, respectively that are closest to $\mathbf{P_f}$. Then, the replicated spline $\mathbf{Q_l}(t)$ is computed as

$$\mathbf{Q_l}^T(t) = \mathbf{Q_r}^T(t) + \mathbf{D} \tag{88}$$

where

$$\mathbf{D} = \begin{bmatrix} 0 & \delta \\ 0 & \delta \\ \vdots & \vdots \end{bmatrix} \tag{89}$$

In the above equation, $\mathbf{D}$ has dimensions of $p \times 2$ where p is equal to the number of elements in the unit interval t from Eq. (85).

Since lane boundaries appear parallel on straight portions of highways, we can use spline replication to estimate the missing spline on either side of the vehicle. Spline replication can also be used to estimate the missing spline on curving portions of highways. To elaborate, lane boundaries appear as concentric arcs. These concentric arcs are segments on the circumference of a circle that share the same center with different radii. Assuming the radius R of the circle on which $\mathbf{Q_r}(t)$ lies, then $\mathbf{Q_l}(t)$ lies on the circle of radius $R-\delta$. Based on the American Association of State Highway and Transportation Officials (AASHTO) specification, the minimum radius of curvature is ~ 602 ft for a highway where the speed limit is 40 mph [74]. This radius increases if the speed limit is increased. Moreover, commercial lane detection and LDW systems require that the vehicle is travelling above 40 mph [75]. Therefore, we also specify this speed limit requirement while testing the MCLDW system and assume that the minimum radius of curvature encountered on a curving road is 602 ft. δ corresponds to the distance between the left and the right lane boundaries measured at points that are closest to the front of the vehicle. Thus, $11 \leq \delta \leq 20$ ft. Therefore, $R >> \delta$ and $R - \delta \approx R$. This idea is illustrated in Fig. 160. We also empirically determined that for $R > 400$ ft, the curvatures of arcs of circles with radius $R \pm \delta$ are nearly identical to the curvature of an arc of a circle with radius R. As a result, we are able to use spline replication to estimate the missing spline.

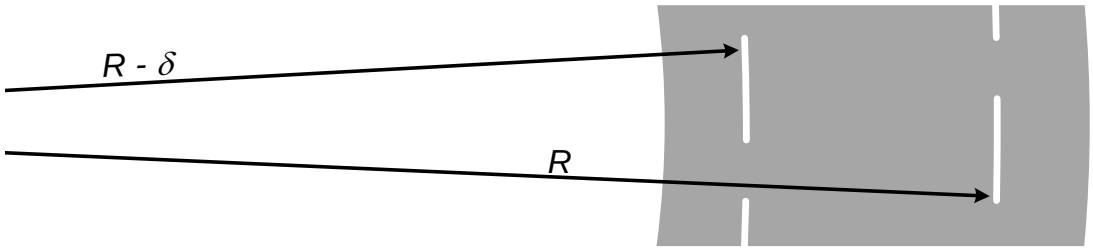


Figure 160: Lane boundaries as concentric arcs separated by a distance δ.

Let us consider another example where only the left lane boundary is detected in $I_{r,n}$. In this situation, a spline $\mathbf{Q_l}(t)$ is created to estimate the left lane boundary outside of the camera views. However, the absence of the right lane boundary in $I_{r,n}$ prevents the creation of the spline $\mathbf{Q_r}(t)$. To estimate the spline $\mathbf{Q_r}(t)$ for right lane boundary, we rearrange Eq. (88) such that

$$\mathbf{Q_r}^T(t) = \mathbf{Q_l}^T(t) - \mathbf{D} \tag{90}$$

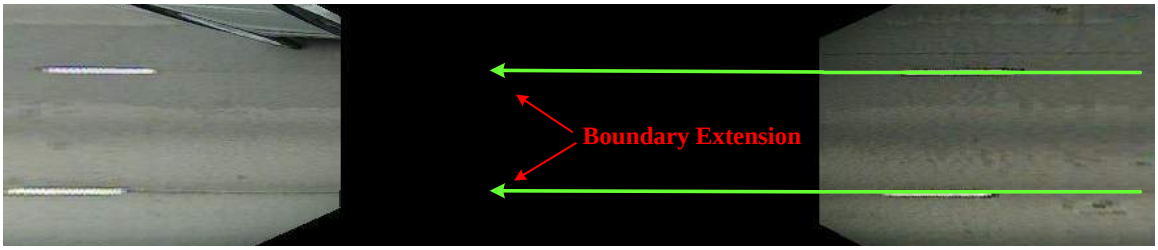
and repeat the process described above. Finally, the spline replication process can also be applied in the case that both lane boundaries are detected in $I_{r,n}$ but either the left or the right lane boundary is not detected in $I_{f,n}$. After modifying δ such that

$$\delta = \begin{bmatrix} \mathbf{Q_{l,3}}^T - \mathbf{Q_{r,3}}^T \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{91}$$

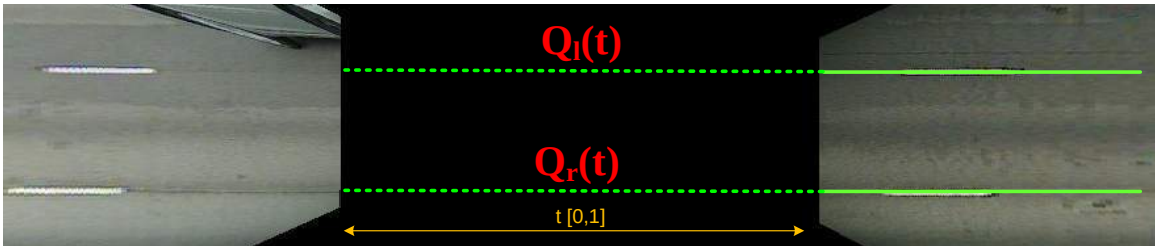
the process to estimate the spline is repeated again. In Eq. (91), $\mathbf{Q_{l,3}}$ and $\mathbf{Q_{r,3}}$ are the two points in $\mathbf{R_{g,l}}$ and $\mathbf{R_{g,r}}$, respectively that are closest to $\mathbf{P_r}$.

Boundary extension is another process of estimating the lane boundaries in the areas that are outside of the camera views by means of extrapolation. To explain in more detail, let us consider an example where the lane boundaries are only detected in $I_{f,n}$. Since no lane boundaries are detected $I_{r,n}$, spline replication cannot be performed to estimate the left or the right lane boundaries. Therefore, extrapolation needs to be performed. This is done by modeling each detected boundary as a straight line and then "extending" this line over a desired length. An example of boundary extension is shown in Fig. 161 where Fig. 161a shows the detected left and right lane boundaries in green. Using the detected lane boundaries as straight line models, the lines are extended over a distance as shown in Fig. 161b to produce the final estimates as shown in Fig. 161c. As with spline replication, the extrapolated lane boundaries are labeled $\mathbf{Q_l}(t)$ and $\mathbf{Q_r}(t)$ for the left and the right lane boundaries, respectively. Here, $t \in [0, 1]$ is the normalized unit interval that corresponds to the distance between

(a) Missing data from rear camera.



(b) Lane boundaries are extended towards the rear.



(c) $\mathbf{Q_l}(t)$ and $\mathbf{Q_r}(t)$ are created.

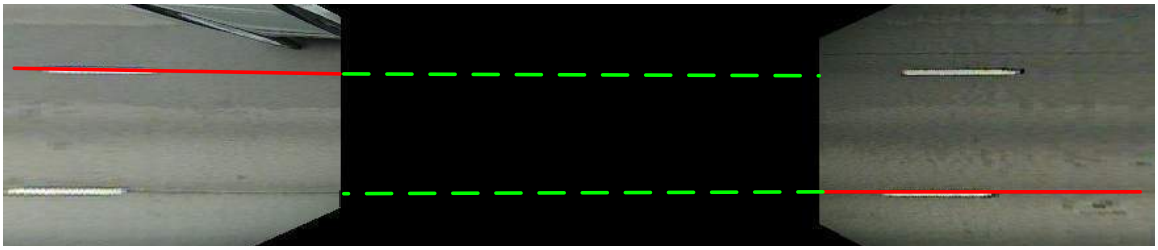
Figure 161: Boundary extension process.

$W_{f,n}$ and $W_{r,n}$ in $W_{g,n}$.

Since the lane detector used in this chapter produces straight line estimates for the lane boundaries, it is to easy to extrapolate the estimates over a desired length. In the case that the lane detector produces curved estimates for the lane boundaries e.g. ALD 2.0, we choose the two end points of the estimates and model a straight line. We use a straight line model because we empirically determined that curved models produced undesirable shapes when extrapolated.

Boundary extension can be similarly used in other scenarios where the lane boundaries are only detected in $I_{r,n}$. Another situation is when the left lane boundary is detected in $I_{f,n}$ and the right lane boundary is detected in $I_{r,n}$ or vice versa as shown in Fig. 162a-162b. In the situation that only the right lane boundary is detected in

(a) Missing data from both cameras.



(b) Lane boundaries are extended in both directions to create $\mathbf{Q_l}(t)$ and $\mathbf{Q_r}(t)$.

Figure 162: Another example of the boundary extension process.

both $I_{f,n}$ and $I_{r,n}$, then the absence of any left lane data prevents the use of either curve replication or boundary extension to estimate the left boundary.

3.5.5 Lane Departure Warning

In a Lane Departure Warning (LDW) system, the driver of the vehicle is informed when the vehicle is within a certain distance from the lane boundary. To perform LDW, we estimate the distance between the vehicle and lane boundaries on either side. As mentioned in Sec. 3.3, existing CCP based LDW systems appear to estimate this distance about 10 ft ahead of the vehicle. In the MCLDW system, the immediate distance between the vehicle and the lane boundary is measured. More specifically, the distance is measured along the y axis of $\mathbf{P_o}$. The distance between the left lane boundary and the vehicle is estimated as

$$\omega_l = \mathbf{Q_l}^T(t_{x=0}) \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{W}{2} \tag{92}$$

where W is the width of the vehicle. In the above equation, $\mathbf{Q_l}(t_{x=0})$ is the evaluation of the $\mathbf{Q_l}(t)$ at a specific value of t such that the x co-ordinate of $\mathbf{Q_l}(t)$ is zero.

Similarly, the distance between the right lane boundary and the vehicle is estimated as

$$\omega_r = -\frac{W}{2} - \mathbf{Q_r}^T(t_{x=0}) \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} \tag{93}$$

The MCLDW system produces a warning to inform the driver of a lane change if the vehicle less than 1 ft away from either boundary. That is, the driver is notified of a lane change if $\omega_l \leq 1$ or $\omega_r \leq 1$.

Fig. 163 provides an illustration of how ω_l and ω_r are computed. In the image,

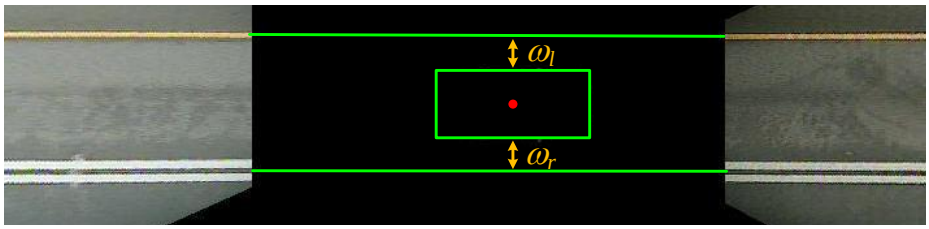


Figure 163: Distance to lane boundaries on either side.

$\mathbf{P_o}$ is shown as the red dot and the green bounding box portrays the dimensions of the reference vehicle in the global co-ordinate system. For visualization, in Fig. 164, we have placed a scaled image of the vehicle in $W_{g,n}$ and shown the estimated lane boundaries on either side of the vehicle.

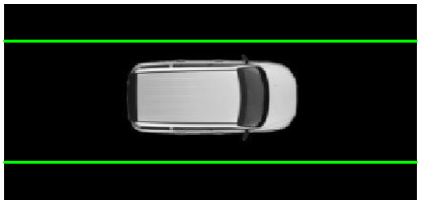


Figure 164: Visualization of the vehicle and estimated lane boundaries.

3.6 Performance Evaluation

To evaluate the MCLDW system, we used five pairs of video clips that were recorded while driving on highways. As described in Sec. 3.4, these clips were recorded using two synchronized cameras that are facing in opposite directions. Testing is performed

only on highway driving footage because commercial LDW systems require that the vehicle is travelling at speed of at least 45 mph [75]. Hence, the MCLDW system is also subjected to the same conditions.

Additionally, testing is only performed on day time driving footage due to the placement of the rear camera. This is because the rear camera captures images of the road through a tinted window. The window tint does not seem to affect the detection of the lane markers in the presence of ambient light. However, at night, the window tint decreases the visibility of the road. As a result, the rear camera captures images that are almost black and the lane markers are nearly invisible. Therefore, the MCLDW system is unable to retrieve useful lane data from the rear camera.

As mentioned in Sec. 3.5.5, the MCLDW system produces a warning if the vehicle is estimated to be less than one foot away from either lane boundary. This warning produced by the MCLDW system is subject to two performance measures:

1. Correct Warning (CW) occurs when both, the MCLDW system's estimate, and the true distance between the vehicle and the lane boundary is less than one foot.

2. False Warning (FW) occurs when the MCLDW system estimates that the vehicle is less than one foot away from the lane boundary; however, the true distance is more than one foot.

Additionally, we also computed the error between the distance estimates produced by the MCLDW system and the true distances. The error for the left lane boundary is computed as

$$\phi_l(n) = |T_l - \omega_l| \tag{94}$$

where T_l is the true distance between the vehicle and the lane boundary, while ω_l is the estimated distance described in Sec. 3.5.5. Similarly, the error for the right lane

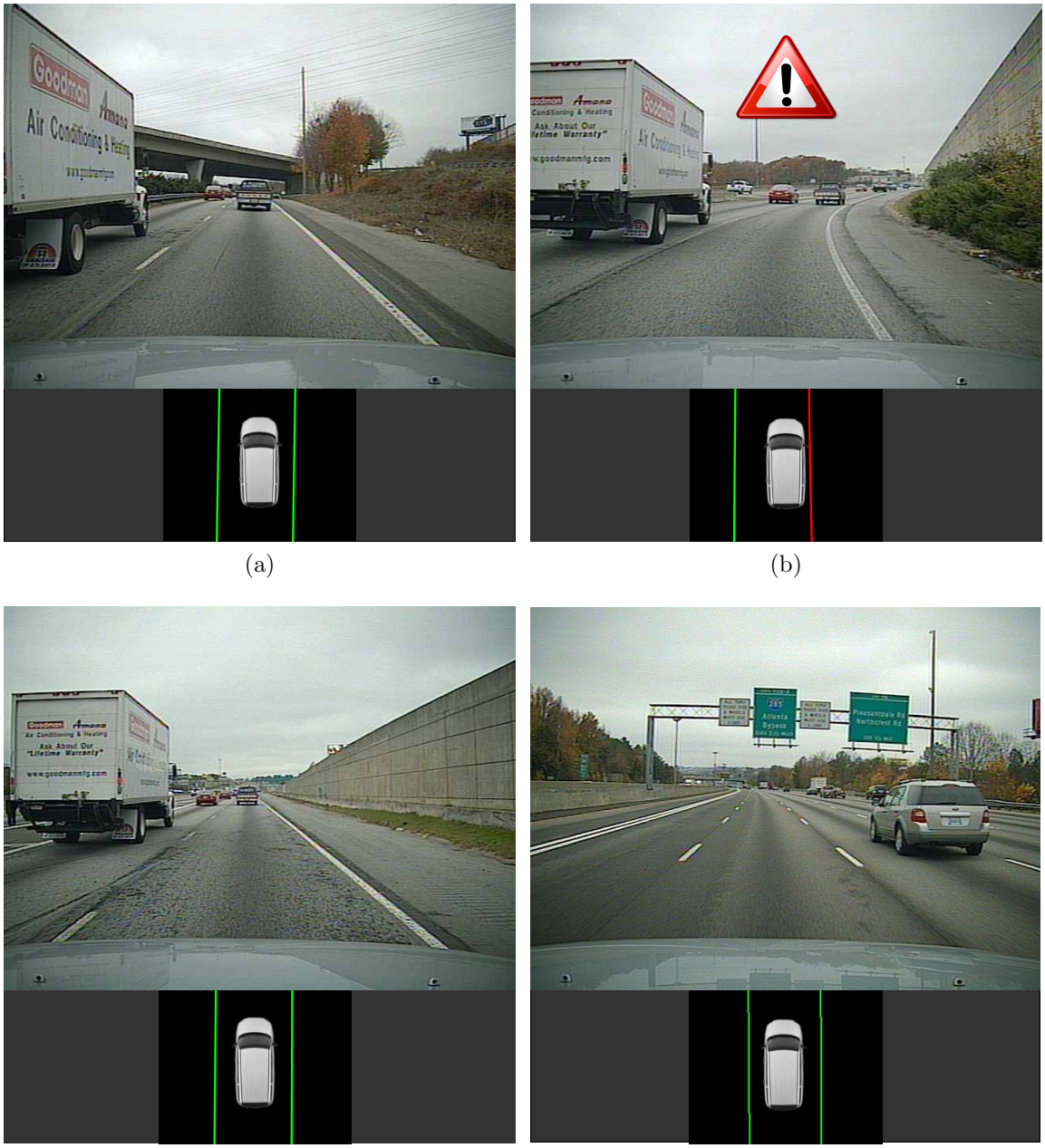
boundary is computed as

$$\phi_r(n) = |T_r - \omega_r| \tag{95}$$

It should be noted that T_l and T_r are actually estimates of the true distances. Since we do not have images of the road that are taken from the sides of the vehicle, we were unable to determine the true distances. Therefore, to determine T_l and T_r, we used the ground truth (see ground truth in Chap. 4) from $I_{f,n}$ and $I_{r,n}$ to produce the Catmull-Rom splines $\mathbf{Q_l}(t)$ and $\mathbf{Q_r}(t)$. Both splines are then evaluated at $t_{x=0}$ as described in Sec. 3.5.5, to determine the estimates of the true distances. Lastly, $\phi_l(n)$ and $\phi_r(n)$ are concatenated to produce $\phi(n)$ which contains the overall error in the clip.

Of the five pairs of video clips used in testing, Clip 1 and Clip 3 contain footage of driving on straight portions of highways. Clip 2 and Clip 4 contain footage of driving on curving portions. Lastly, Clip 5 contains a mix of both straight and curving portions of the highway. In Fig. 165, the performance of the MCLDW system on a few images of these clips is shown. Here, the MCLDW system is in its default mode of operation and utilizes lane data from both cameras to produce estimates of the lane boundaries alongside the vehicle. As shown in Fig. 165b, the MCLDW system successfully indicates a warning when the vehicle is estimated to be under the minimum distance from the lane boundary. In addition, the quantitative performance of the MCLDW system tested on these five pairs of video clips is reported in Table 19. Besides CW and FW, we also computed $\sigma_{\phi(n)}$, the standard deviation of the $\phi(n)$ for each clip. Furthermore, the histogram of $\phi(n)$ for each clip is provided in Appendix D.

In the table, we also provided the performance of the MCLDW system when it uses spline replication and boundary extension to estimate the lane boundaries due to missing data. However, both the left and the right lane boundaries are detected in both $I_{f,n}$ and $I_{r,n}$, respectively, in all clips. Therefore, to test spline replication,



(a) (b)

(c) (d)

Figure 165: Images from the video clips along with a visualization of the vehicle and lane boundary estimates.

the left lane boundary in $I_{r,n}$ is assumed to be undetected. As a result, spline replication produces the spline $\mathbf{Q_l}(t)$ by replicating and offsetting $\mathbf{Q_r}(t)$. Similarly, to test boundary extension, both lane boundaries in $I_{r,n}$ are assumed to be undetected. Therefore, boundary extension extends the lane boundaries in $I_{f,n}$ to produce $\mathbf{Q_l}(t)$

Table 19: Comparison between the different modes of the MCLDW system.

Video Clip	MCLDW Mode	CW %	FW %	$\sigma_{\phi(n)}$
Clip 1	Default	0.0	0.0	0.17
	Spline Replication	0.0	0.0	0.14
	Boundary Extension / CCP	0.0	0.0	0.25
Clip 2	Default	0.0	0.0	0.20
	Spline Replication	0.0	0.0	0.14
	Boundary Extension / CCP	0.0	0.0	0.28
Clip 3	Default	83.33	4.17	0.20
	Spline Replication	83.33	10.31	0.25
	Boundary Extension / CCP	100.00	8.92	0.24
Clip 4	Default	93.45	0.0	0.09
	Spline Replication	93.45	5.32	0.11
	Boundary Extension / CCP	84.11	15.97	0.29
Clip 5	Default	97.59	2.38	0.70
	Spline Replication	97.59	0.32	0.12
	Boundary Extension / CCP	100.00	8.24	0.59

and $\mathbf{Q_r}(t)$. We also intended to compare the MCLDW system to a single camera CCP based LDW system. The CCP approach is modified such that the measured distance is the immediate distance between the vehicle and lane boundary, and not the distance that is measured 10 ft ahead. Fortunately, in this situation, the MCLDW system using boundary extension is identical to the CCP approach. This is because vehicle's position in the lane is estimated based on the images captured by only a single camera.

In Table 19, the MCLDW system produces high rates for correct warnings and low rates for false warning when all boundaries are detected. It also produces similar rates when using spline replication. In addition, $\sigma_{\phi(n)}$ values during both modes of operation are mostly less than 0.25 ft in all clips. This suggests that the distance estimates are close the true estimates at most times. On the other hand, the MCLDW system using boundary extensions also produces high rates for correct warnings, but it also produces much higher rates for false warnings compared to the other two modes of the MCLDW system. The high false warning rates suggest that the distance

estimates are not as accurate as the other two modes. This is also supported by the relatively larger values of $\sigma_{\phi(n)}$ compared to the other two modes in nearly all clips. In Clip 1 and Clip 2, the vehicle is mostly near the middle of the lane. As a result, the distances estimated by all three modes are more than one foot at all times. Hence, neither correct nor false warnings are produced. However, errors in the distance estimates are still computed and are provided in the $\sigma_{\phi(n)}$ column.

As a result, on the test data, it appears that the MCLDW system is able to produce:

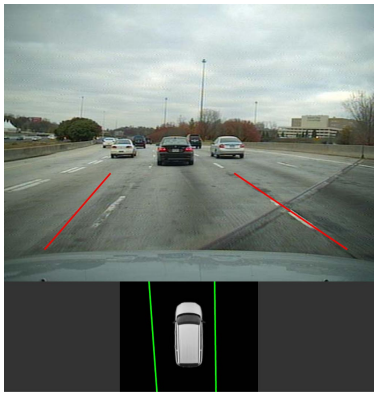
1. A better estimate of the vehicle's position in the lane

2. A better indication of a lane departure

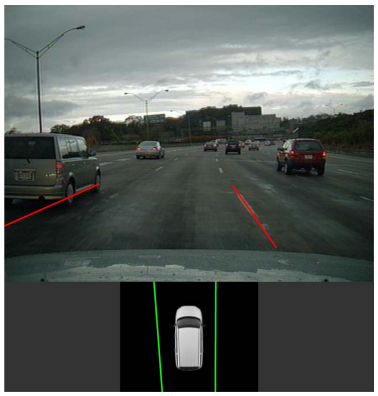
in comparison to the single camera solution such as CCP. However, the MCLDW system also faces some difficulty. Since the MCLDW system relies on the ALD 3.0 to produce accurate lane boundary estimates, errors in the estimated lane boundaries directly reflect on the estimated distance to the lane boundaries. A few examples of misalignments in lane boundary estimates that lead to incorrect lane distance estimates are shown in Fig. 166a-166b. In Fig. 166c, the excessive wear causes the right lane boundary to be undetected in the images of both the front and the rear camera. As a result, spline replication or boundary extensions cannot be performed to estimate the lane boundary.

3.7 Conclusion

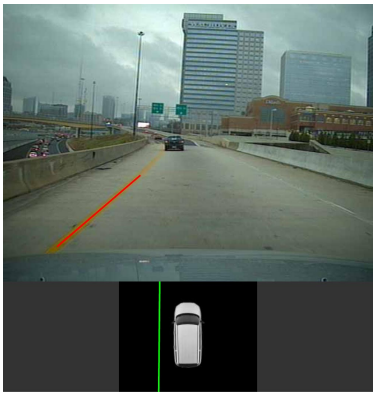
In this chapter, we discussed various details of LDW systems. Following the overview of an LDW system, we provided a brief idea of the MCLDW system that is introduced in this chapter. Then, we specified the requirements for a functioning LDW system and performed a literature survey on the related research in the field that met the requirements. Next, we provided details on how the cameras are calibrated in the



(a) Misalignments in lane boundary estimates.



(b) Misalignments in lane boundary estimates.



(c) Worn out right lane boundary.

Figure 166: A few examples where the MCLDW system faces difficulty.

network. Using an existing lane detector such as the ALD 3.0, lane detection is performed on images captured by both the front and rear cameras. Then, using the detected lane boundaries, Catmull-Rom splines are created. The splines are estimates of the positions of the lane boundaries around the vehicle. Furthermore, these lane boundary estimates are produced in areas that are outside of the cameras fields of view. In the situation that the ALD 3.0 is unable to detect both left and the right lane boundaries, then spline replication or boundary extension are used to produce the desired estimates. The estimates are evaluated at a specific point on the curve to determine the distance between vehicle and the lane boundary. Finally, a warning is produced if the distance is estimated to be less than one foot.

The MCLDW system is tested on real world driving videos and its performance is compared to a CCP based single camera LDW system. The MCLDW system produces high rates for correct warnings and low rates for false warnings. In addition, the MCLDW system also produces small errors in the distance estimates compared to the true estimates in both modes of operation. This is reflected in the histograms of the error $\phi(n)$ in Appendix D as well as the small sigma values for each clip. Besides its good performance, the MCLDW system faces difficulty in situations where lane markers are worn out. In these situations, the lane detector used in the MCLDW

system has difficulty in detecting the lane boundaries; as a result, distance estimates cannot be computed and LDW cannot be performed. One possible solution to address this concern is to incorporate tracking. As with the lane detectors, a tracker could be used to estimate the distance between the vehicle and the lane boundary for a few images even if no measurements are received. Additional improvements will be discussed in Sec. 7.3

The MCLDW system, as with the lane detectors in this dissertation, is implemented in Matlab and operates at about 1 fps on the test bed computer. The bottle necks in its performance are the two lane detectors. Each lane detector operates at about 4 fps, and since the two lane detectors operate asynchronously, the MCLDW system results in a low frame rate. However, the MCLDW system performs simple linear algebra and matrix operations to determine the distance between the vehicle and the lane boundaries. Therefore, to determine its true frame rate, we used lane boundary estimates that were read directly from text files. The data from these files was used as input to the MCLDW system to compute the distance estimates. In this type of experiment, we observed that the MCLDW system is able to operate at about 12 fps. Therefore, if the two lane detectors can be relegated to dedicated hardware, with code optimization, a real time system could be realized.

CHAPTER IV

GROUND TRUTH

4.1 Overview

Performance evaluation involves analyzing and examining the output of a system; as a result, it is an important part of its development. In a lane detection system, the output consists of the estimated locations of the left and right lane boundaries in each image of the video clip. Therefore, performance evaluation of a lane detection system involves analyzing the estimated locations of lane boundaries often with regard to a reference or ground truth. **Ground Truth** for a lane detection system is a dense set of points that define the left and right lane boundary curves in each image of a video clip. An example of lane boundaries defined by the ground truth are shown as the purple curves in Fig. 167. By comparing to ground truth, an objective evaluation of the

Figure 167: Ground truth in an image.

output can be performed. Additionally, the error between the output and reference can also be determined. However, creating the ground truth is a very tedious and time

consuming process; as a result, it is often not created. Consequently, lane detectors are often evaluated on the basis of subjective evaluations. However, subjective evaluation rely on visual inspection which can introduce bias in the results. For example in Fig. 168a, the estimated lane boundary is not overlapping but is very close to the lane

(a) Lane boundary running very close to lane markers.

(b) Lane boundary produces a jagged shape between the gaps.

Figure 168: Estimated lane boundaries.

boundary in the image; thus, one person may consider the estimate as correct while another may consider it to be incorrect. Similarly, in Fig. 168b with dashed lane markers, it difficult to determine if the estimated lane boundary between the gaps

has been estimated correctly. Moreover, it is not possible to determine if any error is present in the output. Based on the discussion above, it is clear that ground truth is necessary for the systematic performance evaluation of a lane detection system. Additionally, finding an efficient way to create ground truth is also equally important.

This chapter focuses on discussing the approaches that are used to create ground truth. Following the introduction, the requirements of ground truth and the approach that is most commonly used to create ground truth will be presented. Next, we will introduce Time-Slicing, a novel approach that allows a user to quickly and efficiently create ground truth. Then, ground truth created using Time-Slicing approach will be compared to reference ground truth for quality. Finally, the details of the output file format will be presented and the chapter will be closed with the conclusion.

4.2 Requirements for the Ground Truth

Since the ground truth describes the curves that represent lane boundaries, the requirements are imposed on the shapes and positions of the curves. These requirements are:

1. The curve representing the lane boundary must be smooth as shown in Fig. 169. The curve must also exhibit no visible kinks in its shape unlike the curve on the right in Fig. 168b.

2. The curve must lie on the center of the lane markers. For double lane markers, it must lie in the gap between the two lane markers as shown in Fig. 169.

3. After specifying an ROI, the curve must start at the top of the ROI and extend towards the bottom of the image along the lane boundary. This idea is illustrated in Fig. 170 where the ROI is shown as a yellow box and the top of the ROI is at a distance of 40 ft ahead of the vehicle. The row number in the image that corresponds to 40 ft can be determined by setting $x = 40$ in Eq. 3. An

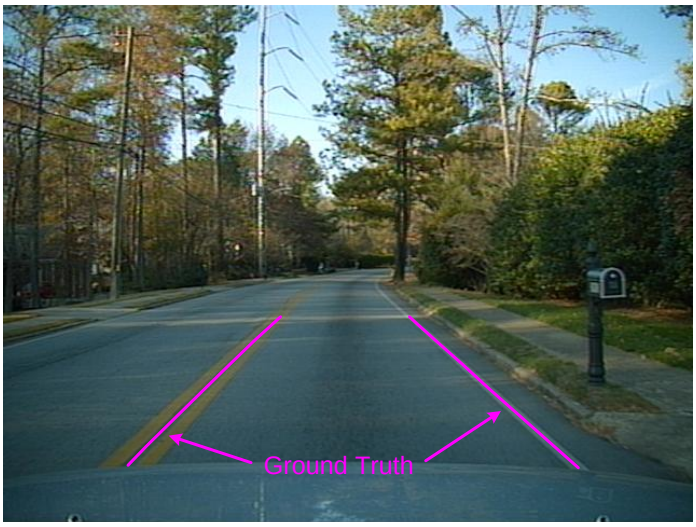


Figure 169: Lane boundary is represented by a smooth curve and lies in the center of the lane markers.

exception to this requirement can occur on sharp turns or where lane markers are out of the camera's view.

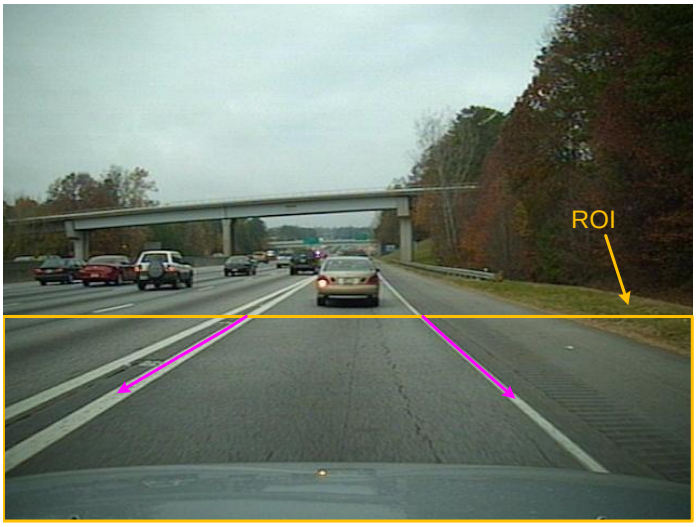


Figure 170: Lane boundary curves (shown as purple arrows) start at the top of the ROI and extend towards the bottom of the image.

4. The lane boundary curve must be provided in the presence of either solid or dashed markers (in between the gaps as well).

4.3 Manual Approach

In this section, we will explain the methodology of the manual approach. This approach is simple, straight forward, and most commonly used to create accurate ground truth. It is performed as follows:

1. The user manually annotates the left lane boundary at a number of points in an image.
2. The gaps between the points are filled in using interpolation.
3. Steps 1 and 2 are similarly performed on the right lane boundary.
4. Steps 1, 2, and 3 are repeated for all images in video clip.
5. Finally, the ground truth is a written to a file.

However, there are two problems with this approach:

1. It is very slow. To annotate a single curve in an image requires meticulous detail and can take up a lot of time. For example, the annotated locations in an image are shown by the pink dots in Fig. 171. Therefore, annotating a large number of images in a video clip can be quite taxing. (The time required to annotate will be discussed in a later section).
2. Gaps between dashed markers are often not annotated because it is difficult to estimate the lane boundary in these areas. This problem is illustrated in Fig. 172.

Additionally, problem #2 affects fulfilling requirement #4 and that can lead to incomplete ground truth. Therefore, ground truth must be created using a methodology that is unaffected by the variations in markers present on the road. Fortunately, the Time-Slicing approach has no such restrictions.

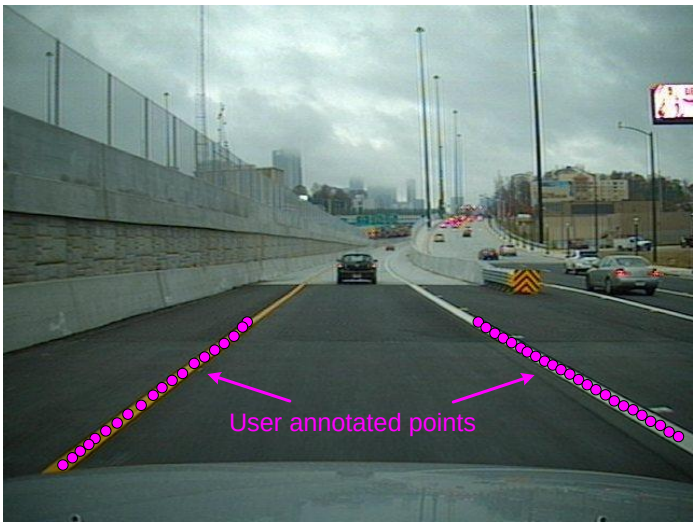


Figure 171: User annotated points in an image.

Figure 172: Difficult to annotate lane boundaries in the gaps.

4.4 Time-Slicing Approach

Time-Slicing is a quick and efficient procedure to create accurate ground truth. One of the advantages of this approach over the manual approach is that Time-Slicing provides us with a spatio-temporal interpretation of the video clip; as result, annotations can be performed quickly and also with good accuracy.

The Time-Slicing approach is broken into 3 stages as shown in the flow diagram in Fig. 173. In Stage 1, the rows to be used for annotation are specified. Then, in

Stage 2, a Time-Sliced image is created for each row and annotations are performed on each Time-Sliced image. Finally, in Stage 3, the annotations from Stage 2 and combined and written out to a file. The details of each stage are provided below:

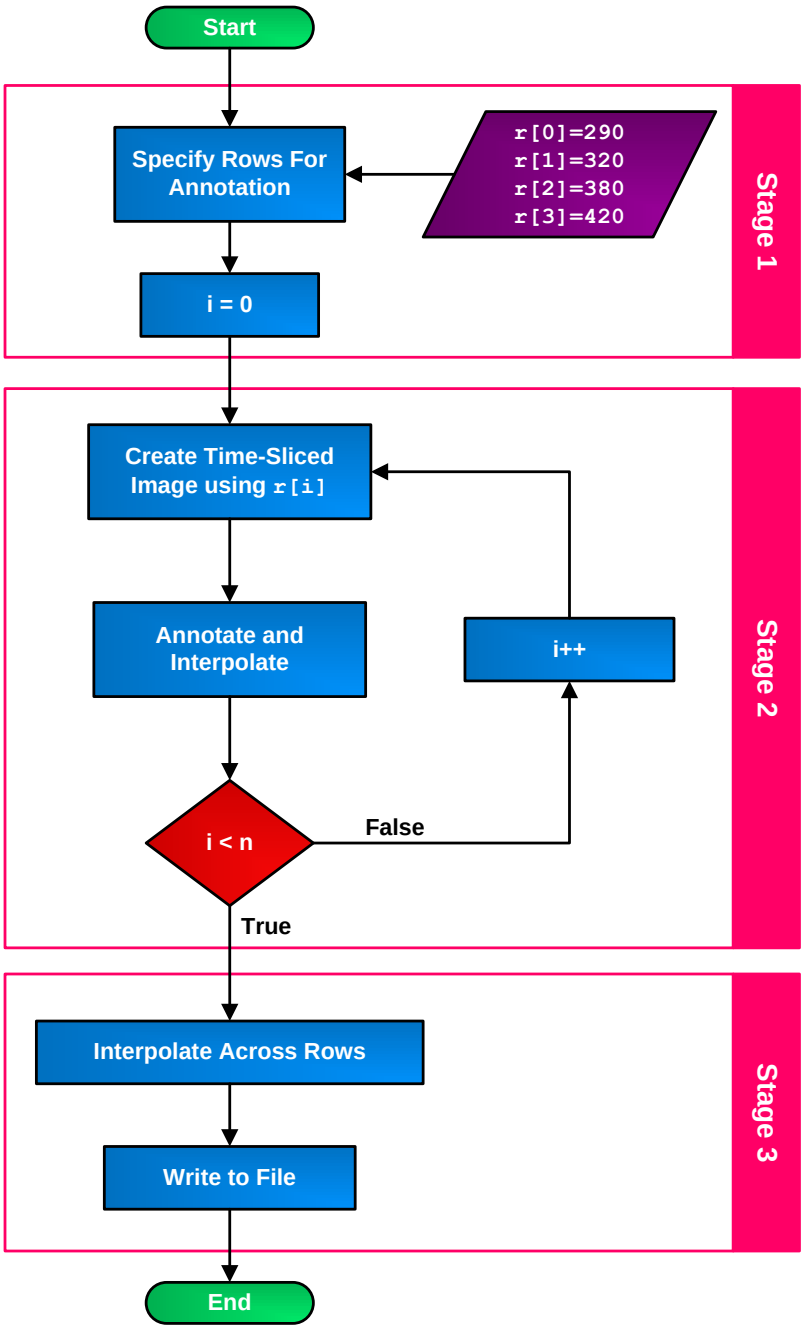


Figure 173: Flow diagram illustrating the mechanics of the Time-Slicing approach.

4.4.1 Stage 1

We first specify a finite number of rows where the lane boundary will be annotated. These rows are shown in Fig. 174. At least three rows must be specified to produce a smooth lane boundary curve. However, we used four rows for this methodology and the specific row numbers are shown in both Fig. 174 and the flow diagram.

Figure 174: The four rows in an image.

4.4.2 Stage 2

This stage consists of two steps that are placed inside a loop. The details of each step are provided below:

1. Creating of the Time-Sliced image: A Time-Sliced image is created by filling an empty image with rows of pixels from images in a video clip. Therefore, the Time-Sliced image that is initially an empty image is created with dimensions $F \times N$, where F is the frame count of the video clip and each image in the video clip has dimensions of $M \times N$. In contrast to the images in the video clip, the Time-Sliced image has dimensions of $F \times N$ because F copies of N pixel wide rows will be placed in the empty image. To simplify the nomenclature in this explanation, row corresponds to a row Time-Sliced image, while `row` (notice the

font change) is a row in an image in the video clip. Next, we specify a `row` in an image from which pixels will be copied. Using the information from Stage 1, let us start with `row=r[0]` (`r[0]=290`). Then, through the use of a loop, a single row of pixels from each image of the video clip is copied to a particular row in the empty image. To elaborate, `row=290` from the first image in the video clip is copied to the first row of the Time-Sliced image. Then, `row=290` from the second image in the video clip is copied to the second row in the Time-Sliced image. Similarly, `row=290` from the F^{th} image is copied to the F^{th} row in the Time-Sliced image. Since the rows of pixel are copied and placed sequentially, they appear to be "stacked" in the Time-Sliced image. The creation of the Time-Sliced image is illustrated in Fig. 175 where rows from the images on the left are copied to the Time-Sliced image on the right. Additionally, a sample Time-Sliced image is shown in Fig. 176. Two observations can be made about this image:

(a) Moving vertically along the rows in the Time-Sliced image is equivalent to traversing through the different images or time stamps in the video clip, due to the "stacking" described earlier

(b) Solid lane markers appear as wavy curves and dashed lane markers appear as small horizontal strips.

Finally, choosing a different value for the `row` e.g. `r[1],r[2]`... will generate a unique Time-Sliced image.

2. Annotation and interpolation: In this step, we first annotate a few points at the centers of the left lane markers in the Time-Sliced image. Then using interpolation, we perform curve fitting to connect the points. Cubic spline interpolation is used here instead of nearest neighbor or linear interpolation because it produces it a smooth curve between the points as shown in Fig. 177.

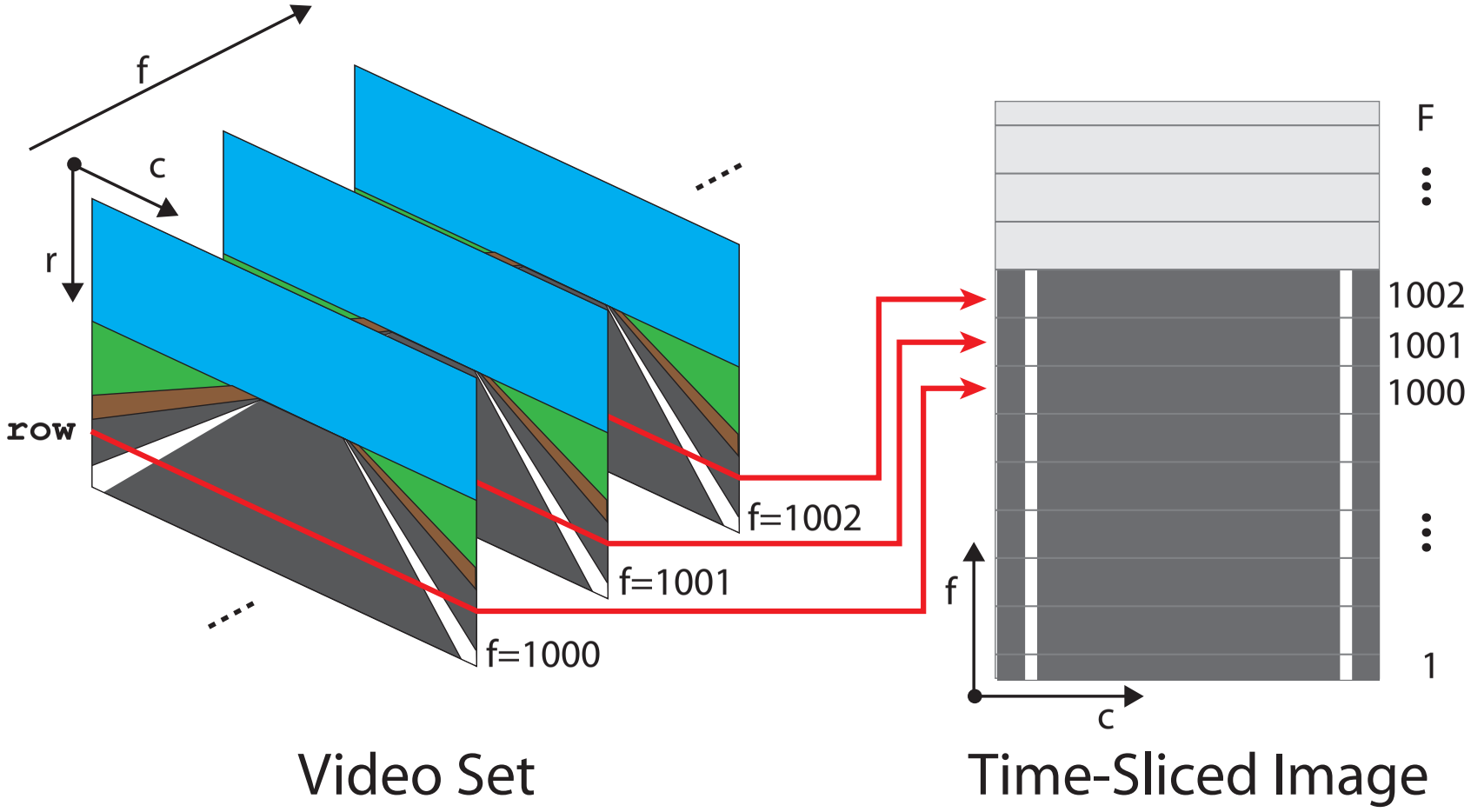


Figure 175: Creating the Time-Sliced image.

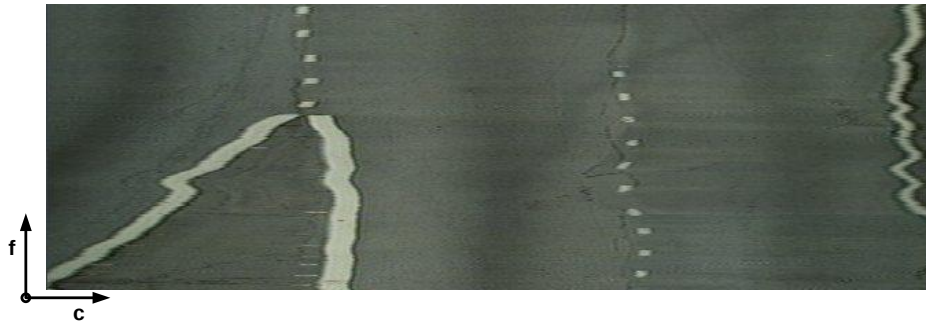


Figure 176: Sample Time-Sliced image with co-ordinate system shown on the bottom left.

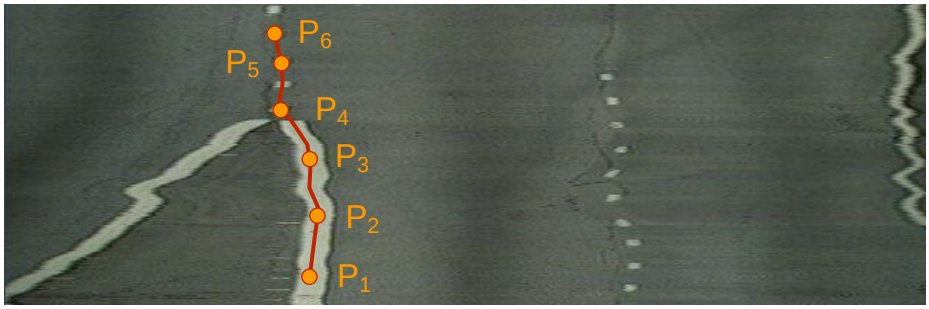


Figure 177: Several annotated points in the Time-Sliced image. Cubic interpolation produces a smooth curve between the points.

Curve fitting serves the purpose of estimating lane marker locations between the annotated points in the Time-Sliced image. To elaborate, consider two points P_1 and P_2 in the Time-Sliced image shown in Fig. 177. P_1 has (f, c) co-ordinates $(10, 35)$ and P_2 has co-ordinates $(20, 40)$. As stated before, a Time-Sliced image is created by stacking a row of pixels from each image in the video clip, in this example `row=290`. Therefore, a value of $f = 10$ in P_1 implies that P_1 is on row 10 of the Time-Sliced image. Moreover, row 10 corresponds to the 10^{th} image in the video clip. Therefore, P_1 in the Time-Sliced image translates to a point on the lane boundary at $(290, 35)$ in image 10 of the video clip. Using this idea, P_2 translates to $(290, 40)$ in image 20 of the video clip. Therefore, lane boundary locations for images $11 - 19$ can be determined by evaluating the curve in the Time-Sliced image on rows $11 - 19$. This idea is illustrated in Fig. 178. Similarly, using interpolation, the locations on the

lane boundaries in other images of the video clip can also be determined. In the same manner, annotating and interpolation are also performed to determine the locations on right lane boundary for `row=290`. Similarly, Time-Sliced images are created for `row=r[1],r[2]`... mentioned in Stage 1. Each Time-Sliced image then undergoes annotation and interpolation described above. Since we have specified four rows in Stage 1, $n = 4$ in Stage 2 of the flow diagram shown in Fig. 173. In Table 20, we have provided the locations for the lane boundary on `r[0],r[1],r[2],` and `r[3]` for the first 20 image in the video clip.

Table 20: Lane boundary locations on the four `rows` in the first 20 image of the video clip.

Image #	Lane boundary location on `row`			
	`r[0]`	`r[1]`	`r[2]`	`r[3]`
1	237.609	196.672	120.071	65
2	236.67	196.396	118.561	63.987
3	235.931	196.125	117.042	63.034
4	235.378	195.861	115.538	62.148
5	234.998	195.611	114.077	61.333
6	234.779	195.377	112.685	60.594
7	234.706	195.166	111.388	59.937
8	234.986	194.976	110.208	59.367
9	235.692	194.811	109.173	58.889
10	236.612	194.679	108.314	58.508
11	237.533	194.588	107.662	58.229
12	238.242	194.549	107.25	58.058
13	238.527	194.554	107.107	58
14	238.478	194.587	107.122	58.055
15	238.336	194.64	107.163	58.198
16	238.115	194.709	107.23	58.394
17	237.827	194.792	107.323	58.606
18	237.484	194.885	107.443	58.802
19	237.099	194.985	107.59	58.945
20	236.684	195.089	107.764	59

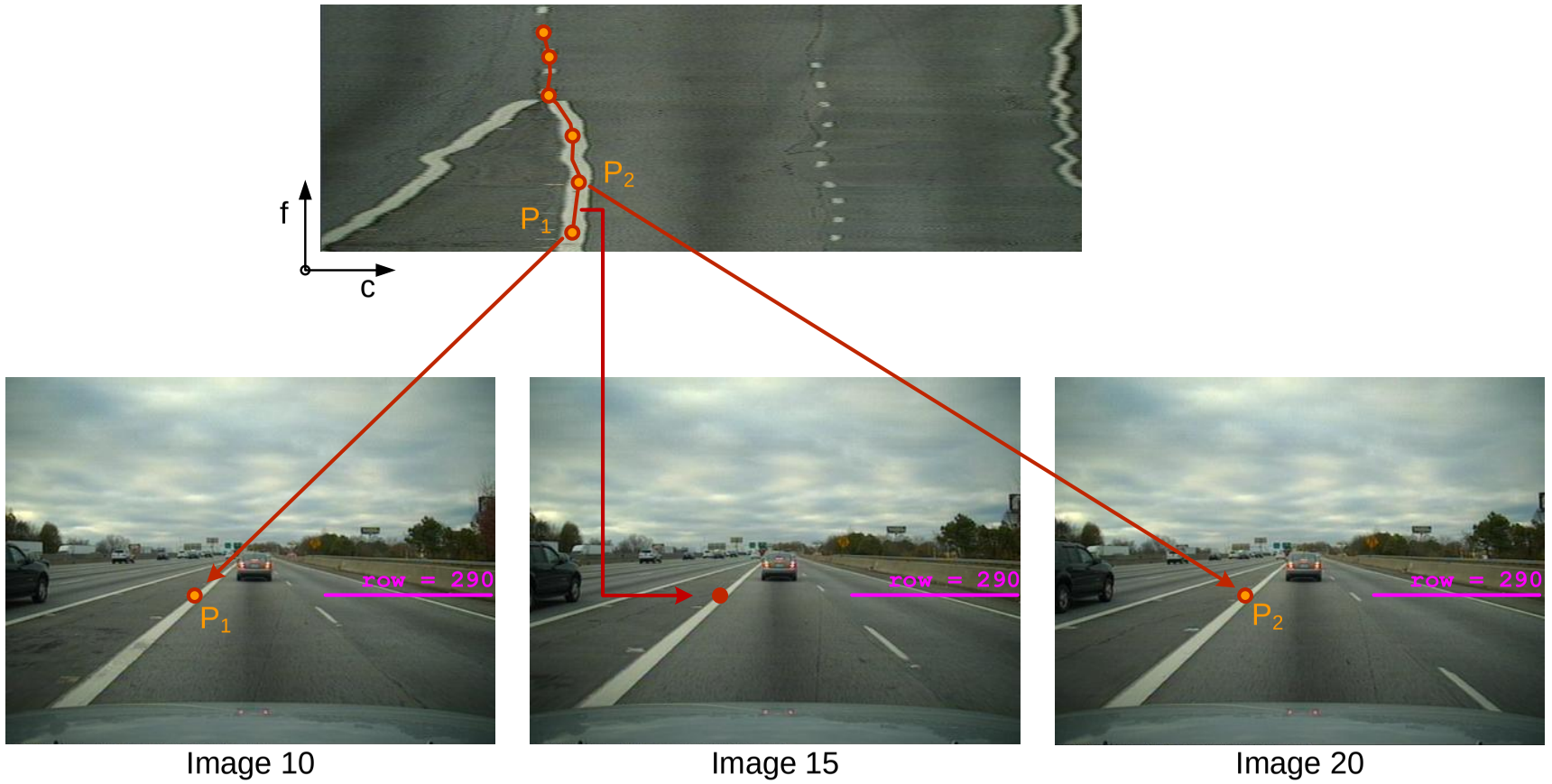


Figure 178: Several annotated points in the Time-Sliced image. Cubic interpolation produces a smooth curve between the points.

4.4.3 Stage 3

At the end of stage 2, we have points of the lane boundaries on four rows in each image of the video clip. However, simply connecting the four locations with straight lines will not produce a smooth curve. Hence, we use interpolation again. But this time, interpolation is performed across the rows in each image as shown. To elaborate, we have four points on the left lane boundary in image 1 as listed on the first row of Table 20. These four points are also shown in Fig. 179. Then, cubic spline

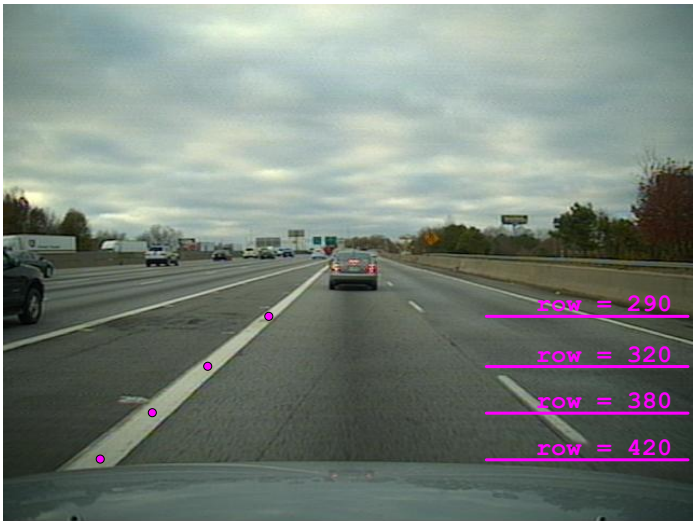


Figure 179: Points on the left lane boundary on the four `rows` specified in Stage 1.

interpolation connects the four rows and at the same time, determines the location of lane boundary on rows $290 - 420$ as shown in Fig. 180. Cubic spline results in a smooth curve for the left lane boundary from `r[0]` to `r[3]` in image 1 as shown in Fig. 181. This process is similarly performed for the right lane boundary in image 1 and then repeated for all the images in the video clip. Table 21 shows the locations of the left and right lane boundaries on rows $290 - 420$ for image 1.

In the end, the lane boundary locations are written to a file in human readable format such as a tab delimited text file or a structured XML file.

Figure 180: Interpolated locations of the lane boundaries between `r[0]` and `r[3]`.

Figure 181: Cubic spline produces a smooth curve for the left lane boundary from `r[0]` to `r[3]`.

4.5 Evaluation and Analysis

Since Time-Slicing relies on a great deal of interpolation to automate and as a result speed up the process to the create ground truth, it is essential to compare its output to ground truth that is created manually in its entirety. For the comparison, we used four video clips that were each 10 seconds in duration. All four clips contain only solid lane markers. Two of these clips were recorded while driving on straight roads, while the other two were recorded on curving roads. Sample images from these clips

Table 21: Lane boundary locations on all `rows` between `r[0]` and `r[3]` in image 1 of the video clip.

Lane boundary location on									
Row	Column	Row	Column	Row	Column	Row	Column	Row	Column
290	237.609	316	202.005	342	168.37	368	135.623	394	101.1
291	236.22	317	200.667	343	167.111	369	134.345	395	99.703
292	234.831	318	199.332	344	165.853	370	133.064	396	98.302
293	233.444	319	198	345	164.596	371	131.781	397	96.895
294	232.057	320	196.672	346	163.341	372	130.494	398	95.483
295	230.672	321	195.349	347	162.085	373	129.204	399	94.067
296	229.288	322	194.03	348	160.831	374	127.911	400	92.646
297	227.905	323	192.716	349	159.577	375	126.614	401	91.221
298	226.524	324	191.405	350	158.323	376	125.313	402	89.792
299	225.145	325	190.099	351	157.069	377	124.009	403	88.358
300	223.767	326	188.797	352	155.815	378	122.701	404	86.921
301	222.391	327	187.499	353	154.561	379	121.388	405	85.48
302	221.016	328	186.204	354	153.307	380	120.071	406	84.035
303	219.643	329	184.912	355	152.052	381	118.75	407	82.587
304	218.273	330	183.624	356	150.796	382	117.424	408	81.136
305	216.904	331	182.339	357	149.54	383	116.093	409	79.681
306	215.538	332	181.057	358	148.282	384	114.757	410	78.224
307	214.173	333	179.778	359	147.024	385	113.417	411	76.763
308	212.811	334	178.502	360	145.764	386	112.071	412	75.301
309	211.451	335	177.228	361	144.503	387	110.719	413	73.835
310	210.094	336	175.957	362	143.24	388	109.363	414	72.368
311	208.739	337	174.688	363	141.976	389	108	415	70.898
312	207.387	338	173.42	364	140.71	390	106.632	416	69.426
313	206.037	339	172.155	365	139.442	391	105.257	417	67.952
314	204.69	340	170.892	366	138.171	392	103.877	418	66.477
315	203.346	341	169.63	367	136.898	393	102.491	419	65.872

are shown in Fig. 182. In the comparison test, we did not use clips that contain dashed lane markers (see problems with manual approach in Sec. 4.3).

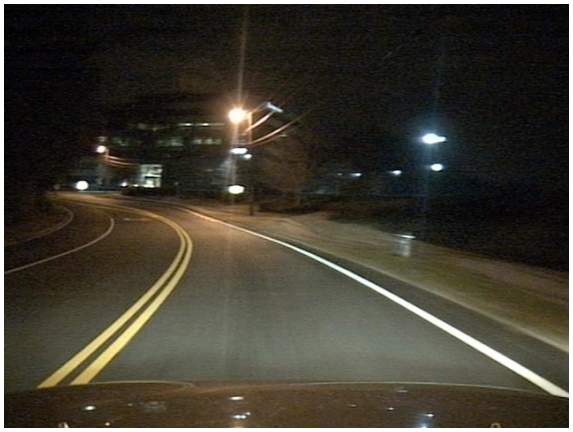
Ground truth is created for each clip using both the manual and Time-Slicing approaches. As in the methodology, four `rows` were used for annotation by the Time-Slicing approach. Then, the error E(f) between the two approaches in each image f is computed as

$$\lambda_{(i,f)} = |Gt_{(i,f)} - X_{(i,f)}| \tag{96}$$

(a) Clip 1.

(b) Clip 2.

(c) Clip 3.



(d) Clip 4.

Figure 182: Sample images from the clips used in the evaluation process.

$$\mathrm{E(f)} = \|\boldsymbol{\lambda}\|_{\infty} = \max_{i} \lambda_{(i,f)} \tag{97}$$

and the performance measure for error is the L_{∞} norm of the λ values. The equations shown above are modified versions of the equations shown in Sec. 2.4.8. Here, Gt represents the ground truth that is created using the manual approach and assumed to be the reference, while X is the ground truth created using the Time-Slicing approach. Additionally, Eq. (96) does not include $\frac{T}{2}$ (the interval around the ground truth); as a result, E(f) computed using the above equations represents the absolute error between the reference and the Time-Slicing approach. Furthermore, $\sigma_{\mathrm{E(f)}}$ is the standard deviation of the distribution of the error E(f) in the clip. Although E(f) is in units of ft as mentioned in Sec. 2.4.8, it is converted to inches for illustration and

is reported in Table. 22.

Table 22: Comparing the manual approach (reference) to the Time-Slicing approach.

Clip	$\sigma_{\mathrm{E(f)}}$	Duration (mins)	
		Manual	Time-Slicing
Clip 1	0.37	59	6
Clip 2	0.401	64	5
Clip 3	0.373	56	8
Clip 4	0.373	51	7

As expected, the ground truth that is created by the Time-Slicing approach is in fact very close to the reference ground truth. This is implied by the small $\sigma_{\mathrm{E(f)}}$ values for each clip in Table 22. In addition, we have also provided the histogram of E(f) for each clip in Fig. 183-186. For illustration, we have also shown some images

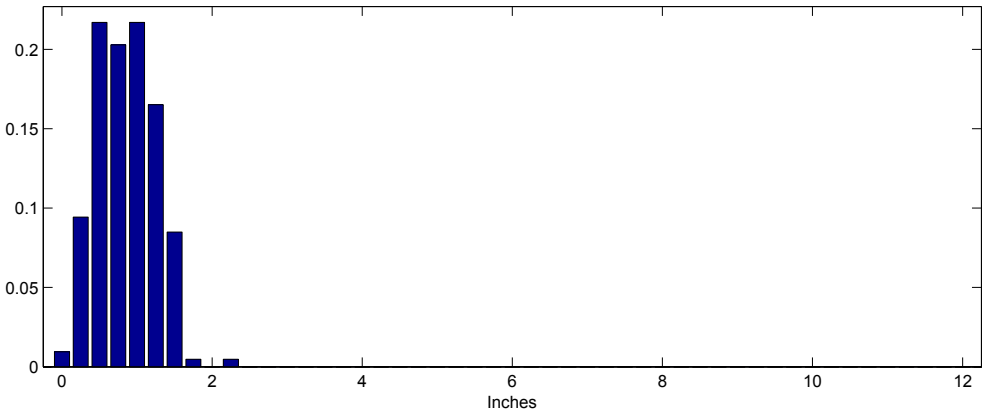


Figure 183: Histogram of E(f) for Clip 1. $\sigma_{\mathrm{E(f)}} = 0.37$

in Fig. 187 where the ground truth that is created manually (green) is overlaid by the ground truth that is created by Time-Slicing (red). However, the lane boundary curves from the two approaches are nearly identical; hence, the curves representing the manual approach are barely visible. As a result, to aid in the illustration, the curves created by Time-Slicing are shown as dashed red curves.

Besides the error, another factor that is an important part of this discussion is the total time required to create the ground truth. Based on the information in Table

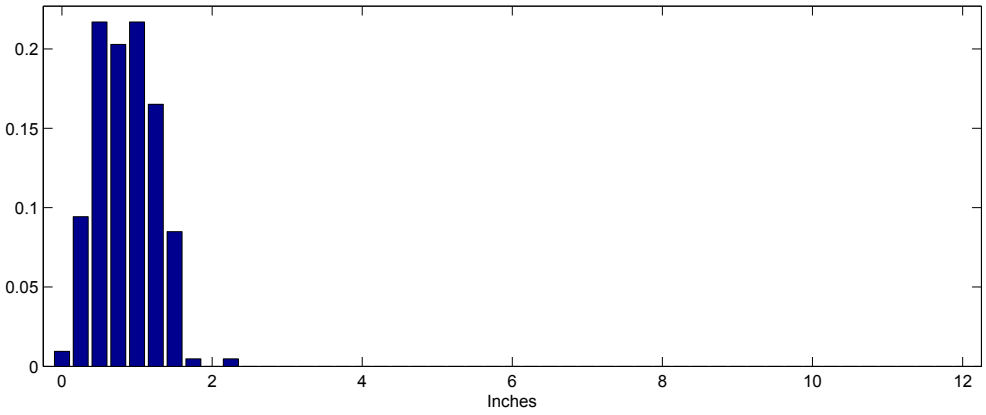


Figure 184: Histogram of E(f) for Clip 2. $\sigma_{\mathrm{E(f)}} = 0.401$

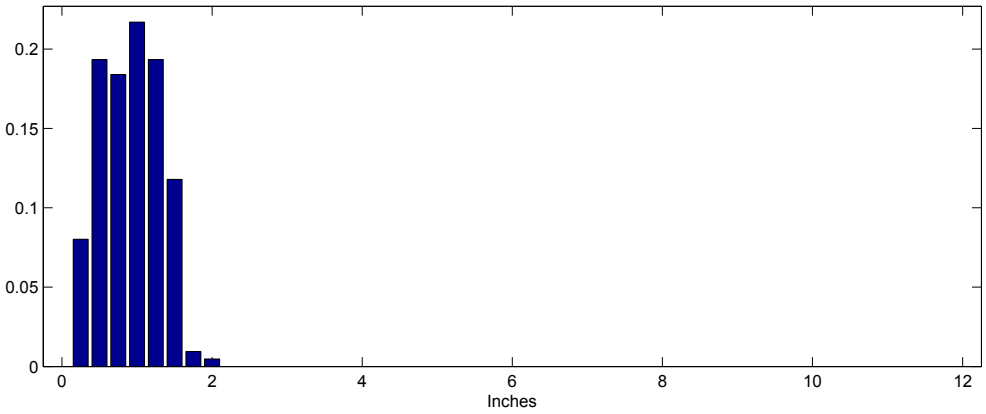


Figure 185: Histogram of E(f) for Clip 3. $\sigma_{\mathrm{E(f)}} = 0.373$

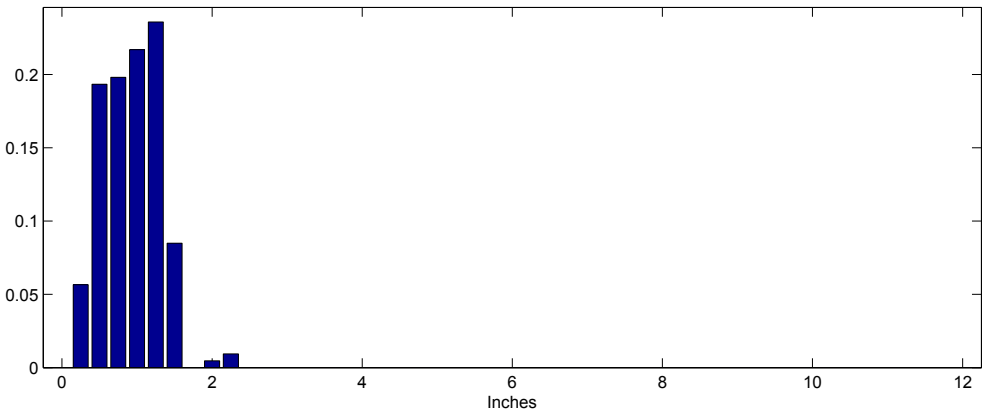
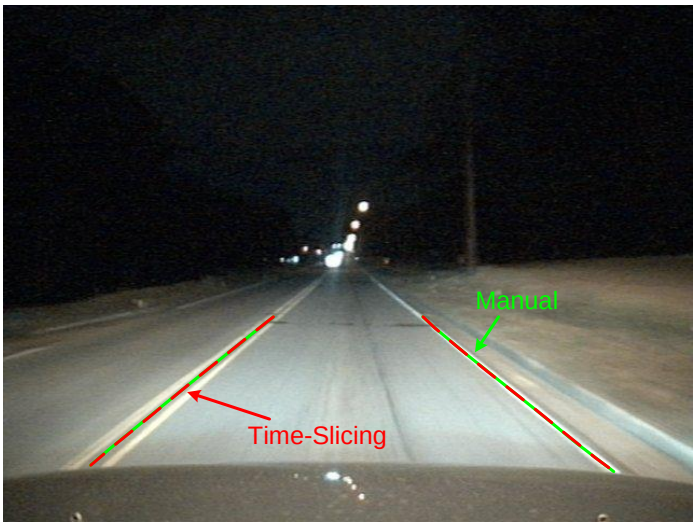
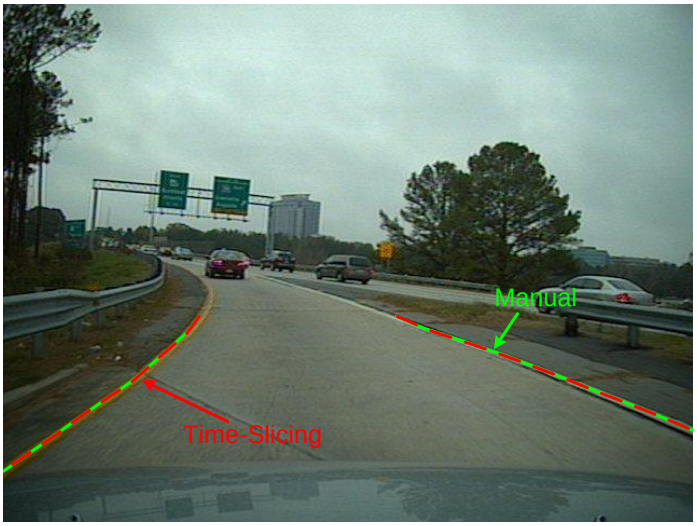


Figure 186: Histogram of E(f) for Clip 4. $\sigma_{\mathrm{E(f)}} = 0.373$



(a) Clip 1.



(b) Clip 3.

Figure 187: Comparison between the ground truth created by the manual approach (red) and Time-Slicing approach (green).

22, creating ground truth using Time-Slicing appears to be much quicker than the manual approach. This was expected because Time-Slicing relies on annotating only a select number of rows in the image and then uses interpolation to fill in the gaps between the selected rows. As a result, each Time-Sliced image took between 30sec to 2 minutes to annotate. Therefore, the total duration to complete the annotation and create ground truth was short. On the other hand, manual approach took anywhere from 30sec to 1minute to annotate a single image. Therefore, when dealing with a

video clip that contains several hundred images, the total duration for this process was much longer.

Finally, based on the clip that was used to explain the mechanics of the Time-Slicing approach, as well as the clips that were used in the analysis above, it is clear the creation of ground truth using Time-Slicing is not restricted to the presence of specific lane markers or specific driving scenarios.

4.6 Output Format

The ground truth is written to a file in XML format. XML or Extensible Markup Language is a simple text-based format that is used for representing structured information [76]. A small snippet from a sample XML ground truth file is provided below.

```
<GroundTruth>
  <Author>Amol Borkar</Author>
  <Institute>GT</Institute>
  <Date>11/4/2010 6:16:21 AM</Date>
  <ID>S1C1</ID>
  <CamNum>0</CamNum>
  <FrameCount>1376</FrameCount>
  <Annotation>
    <Fr ID="1">
      <Left>
        <X>237.609 236.220 234.831 ... 65.872  65.000</X>
        <Y>290.000 291.000 292.000 ... 419.000 420.000</Y>
      </Left>
      <Right>
        <X> ... </X>
        <Y> ... </Y>
      </Right>
    </Fr>
    <Fr ID="2">
      ...
```

Figure 188: Snippet of the ground truth in XML format.

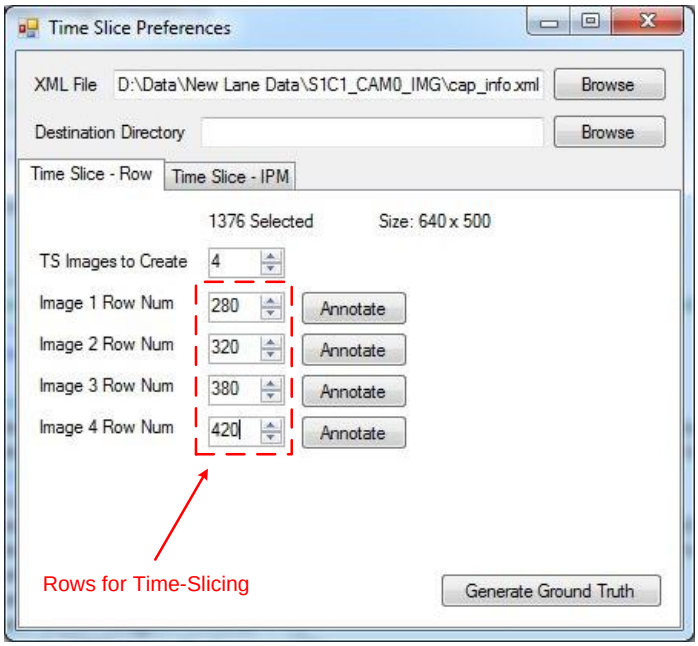
In the XML snippet above, `<Fr>` is the frame element and `ID` is its attribute that

corresponds to the frame number. So <Fr ID="1"> corresponds to frame 1 or image 1 of the video clip. Each frame element has a <Left> and <Right> child; furthermore, both <Left> and <Right> have a pair of child elements named <X> and <Y> that contain the points that define the lane boundaries. For example, the values from Table 21 populate <X> and <Y> in the XML snippet.

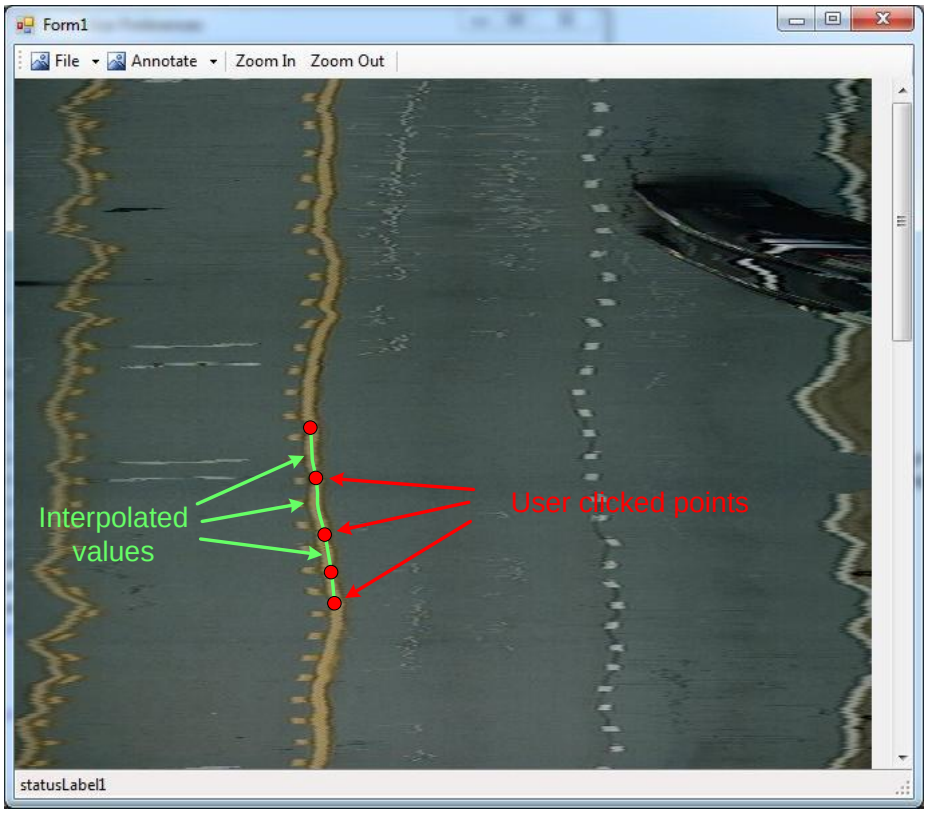
4.7 Conclusion

This chapter is focused on covering various details of the ground truth. Following the introduction, the requirements for the ground truth are specified. Next, the manual approach that is most commonly used to create ground truth is discussed. Then, the Time-Slicing approach is introduced and its methodology is explained through an example. Finally, the ground truth created by both approaches is compared and their performance measures are discussed.

The tool used to create ground truth using the Time-Slicing approach is developed entirely in Visual C#. The tool features a Graphical User Interface that allows the user to select the number of rows to be used for Time-Slicing as shown in Fig. 189a. Then, the user can annotate each Time-Sliced image through mouse clicks as shown in Fig. 189b. The output of the tool is an XML file containing the ground truth in the format described in Sec. 4.6.



(a) User selects rows for Time-Slicing.



(b) Time-Sliced image created for annotation.

Figure 189: Annotation using the tool developed in Visual C#.

CHAPTER V

DATABASE

5.1 Overview

From the discussions covered in the previous chapters, it is clear that the development of lane detection based DA systems is still of much interest to the automotive community. However, due to the large commercial involvement in this field, access to common resources is extremely limited. One of the most important resources is data that is used for evaluation and testing. Comparing to research areas such as face detection or Optical Character Recognition (OCR) where labeled and standardized data sets are often available for training and testing [77, 78], lane detection appears to have no such resources. The problem caused by the lack of common data sets is that existing lane detection algorithms cannot be compared easily. In addition, the lack of common testing data also makes it difficult to validate the performance of other implementations. This was a problem that was discussed in Chapter (lane detection chapter) that made it difficult to choose an appropriate third party lane detector for comparison. The only viable solution in this situation is to implement each algorithm and verify its performance. However, the sheer number of algorithms that have been published in conferences and journals makes it practically impossible to accomplish this task. Therefore, to help in performing fair and unbiased evaluations among the different lane detectors, we have put together a database that is populated with diverse data that was recorded while driving on city roads and highways.

In this chapter, we will first discuss the requirements that are necessary to make a complete database. Next, we will compare some of the existing databases to our requirements. Then, the components that helped formulate our database will be

discussed. Finally, we close the chapter with the conclusion.

5.2 Requirements for a Complete Database

The two components that are necessary to formulate a complete database are the image data and the metadata. The image data must be composed of a collection of image sequences that were recorded while driving at speeds over 20 mph. Furthermore, there should be an interval of at least 2 minutes between the end of one sequence and start of the recording of the next sequence. This also prevents the image data in the database from being a compilation of random images. The requirements for the image data are:

1. Sequence Duration: Each image sequence must be at least 20 seconds in duration. This will allow testing algorithms that use trackers for frame to frame coherence.

2. Color: Images must be captured in color e.g. 24-bit RGB. Since color images can be transformed to grayscale, lane detectors that use either color or grayscale images can use the data.

3. Compression: Images must be stored in an uncompressed or a lossless compression format. This requirement is put in place for two reasons:

   (a) Real-time systems will analyze images from the camera directly; hence, the images in the database need to be representative of the camera's output.

   (b) Lossy compression introduces compression artifacts and affects image quality.

4. Format: Images must be at least in VGA (480×640) resolution to provide good detail of the road surface.

5. Camera Information: Details regarding the cameras intrinsic and extrinsic parameters must be provided to recreate the camera models. The intrinsic parameters include focal length, field of view, and image format. The extrinsic parameters include the yaw, pitch, roll, and the height of the camera's optical center from the ground plane.

6. Lane Markers: Solid and dashed lane markers on straight and curving roads must be present in the image data since they are the most generic lane markers on the roads in the US. However, the image data must also contain images in which lane markers are absent to test for false positives.

7. Road Types: Image data must contain scenes from both city streets and highways as lane detectors tend to be operated on both types of roads. No unpaved or gravel roads.

8. The image data must contain variations in:

   (a) Illumination: Scenes that reflect changing illumination such as driving through tunnels, shadows, under bridges etc. Also, data should be recorded during day, night, and twilight to reflect change in the ambient light levels.

   (b) Weather: Scenes with different weather conditions such as dry, rain, and fog.

   (c) Traffic: Scenes with and without vehicles that are either stationary (parked) or non-stationary (moving) in neighboring lanes.

   (d) Road Surface Textures: Scenes with variety of road surface textures that reflect different contrast levels with respect to the lane markers.

9. Total Duration: The combination of all the sequences should exceed 10 minutes. This duration depicts variety in the sequences present in the database.

Next, we take a look at the metadata. Metadata is the supporting data that acts as a companion to each image in the database. The metadata must contain:

1. Ground Truth: The set of points that describe the left and right lane boundary curves (see definition of ground truth in Chap. 4) must be provided for each image where lane markers are visible. This is one of the most important requirements to enable objective quantification.

2. Vehicle Information (Speed, steering angle, yaw rate, or GPS co-ordinates): One or more of these details must be recorded with each image for estimating ego motion.

5.3 Related Work

In this section, we will take a look at some of the databases that are available online and have been previously used to evaluate lane detectors. Veit et al. [79] used the ROMA database for evaluating a variety of feature extractors for road markings. The ROMA database provides high quality color images with camera information and ground truth for the lane markers. However, the total duration of its image content is less than 20 seconds. Furthermore, the database appears to be compiled from a collection of random images which would make it unsuitable to test lane detectors that incorporate tracking into their algorithms. Nonetheless, databases containing sequences are also available. For example, Leibe [80], Wang [81], and Brostow et al. [82] recorded sequences while driving on local city roads. Although the image content in the databases shows variations in illumination, road surface texture, and presence of neighboring vehicles, the total duration of the image content is less than 10 minutes. Additionally, no ground truth files are provided. On the other hand, the database created by Aly [3] contains ground truth files. But, the image content in [80, 81, 82, 3] does not appear to contain scenes recorded while driving on highways. However, databases containing highway footage are not hard to come by. For example, the

PETS2001 database [83] contains highway sequences while the database created by Sivaraman et al. [84] contains sequences from both city streets and highways. Unfortunately, both databases have compromised on image quality by application of a lossy compression such as JPEG and MPEG-4. Finally, the databases created by Santos et al. [85], Lim et al. [86], and the EISATS database [87] contain a comprehensive set of images that would be ideal for testing lane detectors. The images in [85, 86, 87] show variations in illumination, traffic, road texture, and lane marker types depicting the conditions that a real lane detector would encounter. In addition, the total duration of the sequences in these databases is over 10 minutes and the camera parameters that are necessary to generate camera models are also provided. However, these databases appear to be created for applications other than lane detection. As a result, files containing ground truth are not available; consequently, objective evaluations cannot be performed. Furthermore, algorithms that test on databases described in [85, 86, 87] tend to quantify their results based on visual inspection and can be biased. Therefore, a database that contains large quantities of image data along with ground truth is necessary to enable systematic objective evaluations of lane detection systems.

A summarized comparison between the different databases is provided in Table 23.

5.4 This Database

The database in this dissertation is formulated with the help of two components: the hardware acquisition platform, and the post-processing procedure. The details of these two components are provided below:

5.4.1 Hardware Acquisition Platform

Two NTSC High Dynamic Range (HDR) cameras are used to capture video while driving. One camera is placed looking out of the front windshield and is able to see the road ahead. The second camera looks out of the rear windshield and sees the road

Table 23: Comparison between the different databases available online.

Name[a]	Sequences Present	Color	Compression	Format[b]	Camera Info Avail	Lane Markers[d]	Road Types[e]	Variations in[c]				Total Dur. (min)	Ground Truth Avail	Vehicle Info Avail
								Illum.	Weather	Traffic	Road Tex.			
[79]	N	Y	JPEG	>VGA	Y	S,D	C,H	Y	Y	Y	Y	<1	Y	N
[80]	Y	Y	PNG	>VGA	Y	S,D	C	Y	N	Y	Y	2	N	N
[81]	Y	Y	PNG	<VGA	Y	S,D	C	Y	Y	Y	Y	2	N	N
[82]	Y	Y	MXF	>VGA	Y	S,D	C	Y	Y	Y	Y	9	Y	N
[3]	Y	Y	PNG	VGA	Y	S,D	C	Y	N	Y	Y	2	Y	N
[83]	Y	Y	JPEG	>VGA	Y	S,D	H	Y	N	Y	N	2	N	N
[84]	Y	Y	MPEG-4	>VGA	N	S,D	C,H	Y	Y	Y	Y	2	N	N
[85][f]	Y	Y	Unknown	Un	Y	S,D	C,H	Y	Y	Y	Y	51	N	Y
[86]	Y	Y	MPEG-2	>VGA	Y	S,D	C,H	Y	Y	Y	Y	60	N	N
[87]	Y	Y	PPM	VGA	Y	S,D	C,H	Y	Y	Y	Y	12	N	Y
This	Y	Y	PNG	VGA	Y	S,D	C,H	Y	Y	Y	Y	98	Y	Y

[a]Referred to by citation except for the last database.
[b]>VGA: Higher than VGA, <VGA: Lower than VGA, Un:Unknown.
[c]Only checked for variances in the subcategories. Specific variations e.g. shadows, dry, wet, asphalt etc. are not listed.
[d]S:Solid Markers, D:Dashed Markers.
[e]C:City, H:Highway.
[f]Images are stored in a proprietary format. No response from authors. Assessments made based on sample images and descriptions of datasets.

behind. Unlike CCD cameras, HDR cameras are capable of producing high quality images that more accurately represent the range of intensity levels found in real scenes, ranging from direct sunlight to faint starlight [88]; therefore, HDR cameras are used in the acquisition platform. Fig. 190 illustrates the difference in images captured by a CCD camera and an HDR camera. The cameras interface with a computer through

(a) Image from a CCD camera. (b) Image from an HDR camera.

Figure 190: A comparison between images captured by a CCD camera and an HDR camera [7].

a multi-channel frame grabber [89]. Besides video, the speed of the vehicle is also recorded through the aid of an On-Board Diagnostics II (OBD II) reader [90]. The OBD II reader is capable of recovering additional vehicular diagnostic information such as engine temperature, brake pedal position, and revolutions per minute (RPM) etc. Unfortunately, steering angle and yaw rate that are necessary in estimating ego motion are not part of the OBD-II specifications [91]; consequently, it was not possible to recover these parameters.

During a recording session, a timer is initiated and then video from the two cameras are captured simultaneously by the frame grabber. The frame grabber digitizes the video and stores it onto Serial ATA (SATA) hard disks as a sequence of Bitmap (BMP) image files in VGA resolution and 24-bit RGB color. The naming convention for each BMP file is:

`CAP_AA_BBBBBBB_HH.MM.SS.GGG_RRR.bmp` where:

- `AA`: Camera ID number.
- `BBBBBBB`: Zero padded frame number.
- `HH`: Hour component of the timer.
- `MM`: Minute component of the timer.
- `SS`: Seconds component of the time timer.
- `GGG`: Milliseconds component of the timer.
- `RRR`: - Speed of the vehicle at that instant.

So, for example an image with a filename `CAP_00_0000341_00.11.12.321_62.bmp` can be read as an image captured by camera 0, that has a frame number of 341, captured after a duration of 00:11:12.321 since the timer was started, while the vehicle was traveling at 62 mph. The reason for adding the timestamp to the filename is to help in synchronizing the images captured from the two cameras. For instance, if we were using only frame numbers, it would be impossible to synchronize the captured images if one or more "frame drops" were to occur during the recording session.

At the end of each recording session, a file named `cap_info.xml` is created and placed in the folder containing the image sequence. A sample `cap_info.xml` is provided below. This file contains details regarding the duration of the recording session, the number of frames captured, date and time of capture, and the camera's intrinsic and extrinsic parameters. An illustration of the interactions between the different hardware components is shown in Fig. 191. Pictures of the hardware platform are also shown in Fig. 193.

5.5 Post-Processing Procedure

The post-processing procedure consists of three steps that are described below:

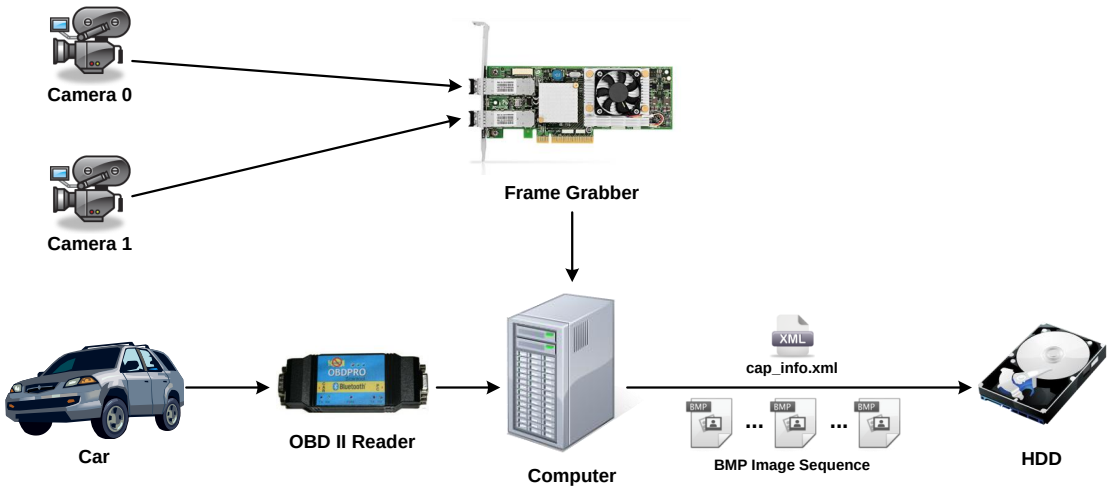


Figure 191: Components of the hardware acquisition platform.

5.5.1 Trimming

Trimming is the process of extracting only relevant segments or clips from the image sequence that was recorded. Trimming is important because the image sequence may contain segments when the vehicle is parked or stopped in the middle of traffic. These segments bear no use in the evaluation of lane detectors and do not need to be in the database. Sample images from such segments are shown in Fig. 194. Using trimming, 1-2 minute duration clips are extracted to temporary folders and forwarded to the next step. Each clip is also accompanied by a modified `cap_info.xml` that reflects a reduced frame count and duration as a result of trimming.

5.5.2 Creating the Ground Truth

Ground truth is created for each clip using the Time-Slicing approach described in Chap. 4. The ground truth is then stored as an XML file named `GT.xml` and placed in the same folder that contains the clip.

```
<CaptureInfo>
  <ID>S1C1</ID>
  <CamNum>0</CamNum>
  <RecordingDate>12/6/2009 8:31:22 PM</RecordingDate>
  <Duration>00:56:58.9908664</Duration>
  <FrameCount>92171</FrameCount>
  <Format>NTSC</Format>
  <FPS>FPS_30</FPS>
  <FrameSize>VGA</FrameSize>
  <FrameGrabberProperties>
    <Manufacuturer>The Imaging Source GmbH</Manufacuturer>
    <ModelInfo>PCIe 4Ch Grabber</ModelInfo>
  </FrameGrabberProperties>
  <CameraProperties>
    <Manufacuturer>Dallmeier Electronics GmbH</Manufacuturer>
    <ModelInfo>MDF3000A-CS-DN</ModelInfo>
    <Height>5</Height>
    <Incline>0.0084</Incline>
    <Yaw>0.0281</Yaw>
    <Roll>0</Roll>
  </CameraProperties>
  <LensProperties>
    <Manufacuturer>Tamron</Manufacuturer>
    <ModelInfo>13VG550ASII-SQ</ModelInfo>
    <FOV_H>52</FOV_H>
    <FOV_V>40</FOV_V>
    <Focal_Length>4.9331</Focal_Length>
  </LensProperties>
</CaptureInfo>
```

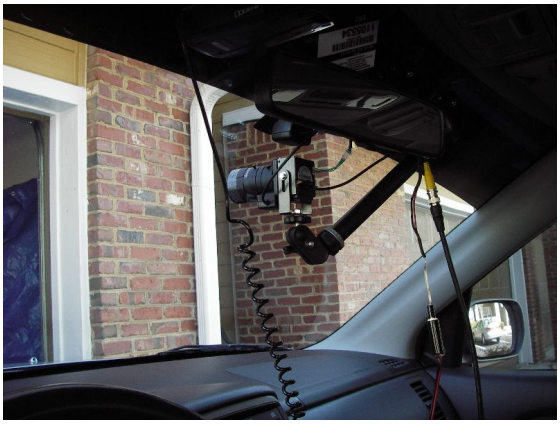
Figure 192: A sample `cap_info.xml` file.

5.5.3 Inspection

In this step, the ground truth of each clip is visually inspected for quality and precision. This is done by overlaying an image with the ground truth and then checking for two criteria:

1. The lane markers (solid or dashed) in each image are completely overlapped by the ground truth.

(a) Computer placed in the trunk.



(b) Front camera.

(c) Rear camera.

Figure 193: Hardware equipment used to record video.

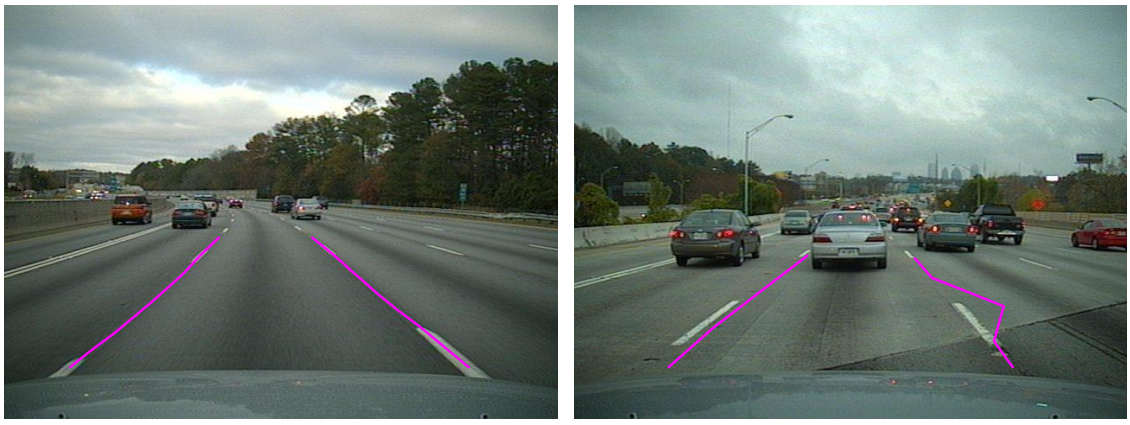
(a) Stopped in a parking deck.

(b) Stopped in the middle of traffic.

Figure 194: Images recorded while the vehicle was stationary.

2. The ground truth is represented as a smooth curve.

If these criteria are not met, then the ground truth in the image need to be corrected. To elaborate, Fig. 195a is an example where the ground truth completely overlaps the lane markers and at the same time also is also represented as a pair of smooth curves. On the other hand, Fig. 195b is an example where the ground truth in the



(a) Image with proper ground truth. (b) Ground truth that needs fixing.

Figure 195: Inspecting the ground truth data.

image need to be corrected. This can be explained by observing the ground truth for the left lane boundary which is represented as a smooth curve but does not overlap the lane marker, while the ground truth for right lane boundary is represented as a jagged curve and only partially overlaps the right lane marker in the image. To prevent bias in inspection, this task is performed by another user. Upon completing the inspection, each image in the clip is converted from BMP to PNG format. PNG or Portable Network Graphics [92] is a lossless image compression format that reduces the storage requirements of the image without affecting its quality. Finally, the PNG images, `cap_info.xml`, and `GT.xml` are compressed in a RAR (Roshal ARchive) [93] file and then uploaded to the server.

An illustration of the post-processing procedure described above is provided in Fig. 196.

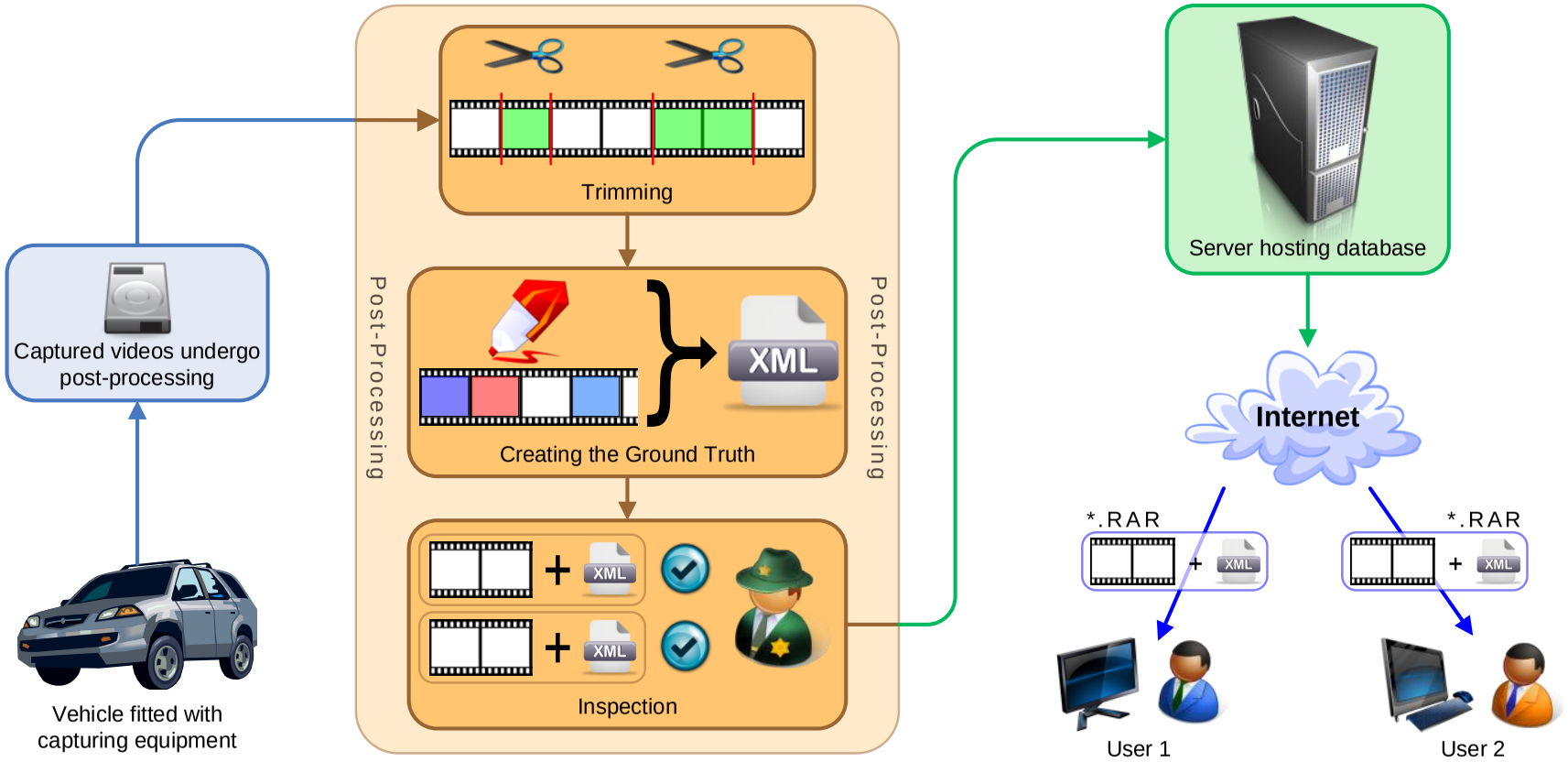


Figure 196: Post-Processing Procedure.

5.6 Access and Availability

The database is free to use and is located at:

`ftp://ftpx.ece.gatech.edu/Db_Mhh_Aab/LaneData`

Login information can be acquired by contacting the authors. At the moment, the database contains 78 clips that are on average 2 minutes in duration each and stored in separate RAR archives. Additionally, each archive has its own companion preview video file. The video file serves the purpose of providing the user with a "preview" of the contents in the clip without having to download the archive since each archive is approximately 2GB in size.

Numerous recording sessions were performed during various times of the day and night, on different road surfaces, during diverse weather conditions, and in the presence of varying traffic volumes to create a very realistic set of conditions that a lane detector would encounter. Sample images that represent the content in the database are shown in Fig. 197. Based on the diversity of content in the image data, and the availability of the metadata, it is clear that the database presented in this chapter is the most complete in regard to the requirements specified in Sec. 5.2. As a result, this database is ideal for evaluating lane detectors as well as lane detection based DA systems. Finally, a comparison between the database presented in this chapter and some of the existing databases discussed in Sec. 5.3 can be found by referring back to Table 23.

5.7 Conclusion

This chapter focused on the development of a database of driving videos that will be used in the evaluation of lane detection based DA systems. Following the overview, we discussed the requirements to make a complete database and then compared a few of the existing databases to our requirements. Next, details regarding the hardware acquisition platform and post-processing procedure that were used to create the

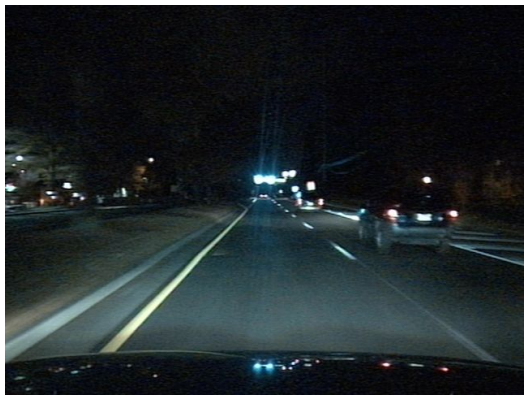
(a) Changing illumination.

(b) Worn out markers.

(c) Shadow conditions.

(d) Navigational information.

(e) Highway.



(f) Night-time driving.

Figure 197: Images in the database captured by the forward facing camera.

database were provided. Finally, we showed some sample images from the database and provided details on how to access it.

CHAPTER VI

PUBLICATIONS

In this chapter, we will summarize the contributions of the publications that formulated the basis of this dissertation. The publications are not listed in chronological order, but in the order of their contributions in the dissertation.

1. A. Borkar, M. Hayes, M. Smith, S. Pankanti. "A Layered Approach To Robust Lane Detection At Night." *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems (CIVVS 2009)*, March 2009: This publication formulates the basis of the ALD 1.0 in Sec. 2.4. In the publication, we introduce the Preprocessing, Object of Interest Detection, Preliminary and Detail Feature Extraction blocks that are documented in this dissertation. The lane detector is qualitatively assessed and its performance is tabulated.

2. A. Borkar, M. Hayes, M. Smith. "Robust Lane Detection And Tracking With Ransac and Kalman Filter." *IEEE International Conference on Image Processing (ICIP 2009)*, November 2009: This publication presents an extension of the work described above. It integrates the Inverse Perspective Mapping (IPM), Data Modelling, and Tracking blocks to the previous design of the ALD 1.0. The addition of the new blocks show improved results.

3. A. Borkar, M. Hayes, M. Smith. "Lane Detection And Tracking Using A Layered Approach." *Advanced Concepts for Intelligent Vision Systems (ACIVS 2009)*, September 2009: In this publication, the ALD 1.0 is extensively tested on real world data. In addition, its performance is compared to the Hough transform and the Middle-to-Side lane detectors.

4. A. Borkar, M. Hayes, M. Smith. "Detecting Lane Markers In Complex Urban Environments." *IEEE Workshop on Intelligent Vehicular Networks (InVeNET 2010)*, November 2010: This publication formulates the basis of the ALD 2.0 in Sec. 2.5. In the publication, we introduce the Filtering, Lane Region Merging, and Lane Marker Classifier blocks that are used by the ALD 2.0. The lane detector is quantitatively assessed and its performance is tabulated.

5. A. Borkar, M. Hayes, M. Smith. "Detecting Lane Markers In Complex Environments Using A Monocular Camera". *IASTED Signal Processing, Pattern Recognition and Applications (SPPRA 2011)*, February 2011: This publication introduces the Piecewise Tracker that is used to estimate the position of the lane boundary in the image. The addition of the tracker, improves the results of the lane detector. Besides the tracker, the publication also introduces the error calculation technique described in Sec. 2.4.8 to objectively evaluate the lane detectors.

6. A. Borkar, M. Hayes, M. Smith. "Advanced Lane Detection Using Elliptical Lane Marker Grouping And Cascaded Templates." *Advanced Concepts for Intelligent Vision Systems (ACIVS 2010)*, December 2010: This publication improves the original design of the Lane Marker Classifier block by performing windowing. This reduces the number of misalignments and false detections that were seen in the previous increment of the ALD 2.0.

7. A. Borkar, M. Hayes, M. Smith. "Lane Detection Using Constraints Of Parallel Lines". *IEEE Conference on Consumer Electronics (ICCE 2011)*, January 2011: This publication formulates the basis of the ALD 3.0 in Sec. 2.6. In this publication, we introduce the Parallel Line Detection block that is used by the ALD 3.0. The lane detector is tested on videos of real world driving conditions and shows good performance.

8. A. Borkar, M. Hayes, M. Smith. "Polar Randomized Hough Transform For Lane Detection Using Loose Constraints of Parallel Lines." *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, May 2011: This publication presents an extension of the work described above. Here, we introduce the Polar Randomized Hough Transform block which acts as an improvement over the conventional Randomized Hough Transform to efficiently find lines in the image.

9. A. Borkar, M. Hayes, M. Smith. "A New Multi-Camera Approach For Lane Departure Warning." *Advanced Concepts for Intelligent Vision Systems (ACIVS 2011)*, August 2011: In this publication, we introduced the idea of the MCLDW system presented in Chap. 3. This publication goes on to explain the camera calibration procedure, Global Co-ordinate System, linear spline, and LDW blocks. The MCLDW system is tested on videos of real world driving conditions and shows good performance.

10. A. Borkar, M. Hayes, M. Smith. "A Non Overlapping Camera Network: Calibration and Application Towards Lane Departure Warning." *International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV 2011)*, July 2011: This publication improves the work presented in the publication above. The improvement replaces the existing linear spline with a Catmull-Rom spline to produce a better estimate of the lane boundaries alongside the vehicle. This gives rises to the Lane Boundary Connection block in the MCLDW system.

11. A. Borkar, M. Hayes, M. Smith. "An Efficient Method to Generate Ground Truth For Evaluating Lane Detection Systems." *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, April 2010: In this publication, we introduce the Time-Slicing technique that is described in

Chap. 4. In addition, we also show how Time-Slicing is used to efficiently create ground truth that is necessary to objectively evaluate the lane detectors.

12. A. Borkar, M. Hayes, M. Smith. "A Novel Lane Detection System with Efficient Ground Truth Generation." *To appear in IEEE Transactions on Intelligent Transportation Systems*: This publication provides extensive implementation details of the ALD 1.0. Besides reiterating the Time-Slicing technique, it also introduces the idea of establishing a common database to compare the lane detectors as described in Chap. 5.

CHAPTER VII

CLOSING REMARKS

7.1 Conclusion

In this dissertation, we have developed a Multi-Camera Lane Departure Warning (MCLDW) system and a framework to evaluate it. Since a lane detector is the core component of an LDW system, we began the dissertation by covering several details of a lane detector. Following the introduction, we specified the requirements that formulate a functioning lane detector. Then, we took a look at some existing research in the field. Next, we presented three new lane detection methodologies. Each methodology contained several new components that were explained in detail in Chap. 2. Each lane detector was then systematically evaluated by comparing its estimates to the ground truth and computing the error. After selecting the best performing lane detector, we proceeded to explain the MCLDW system. Following a brief overview of the idea behind the MCLDW system, we presented the related research in the field. Then, we explained the camera calibration procedure and technique to combine the data from the lane detectors to produce an accurate estimate of the lane boundaries with respect to the vehicle. We also introduced spline replication and boundary extension processes that address the concern of estimating the lane boundaries if one of the two cameras is unable to detect one or more lane boundaries. Using the MCLDW system, we were able to accurate determine the vehicle's position in the lane and the times when it appeared that vehicle was about to change lanes. Then, as part of the evaluation framework, we introduced Time-Slicing. Time-Slicing is a procedure to quickly and efficiently annotate the video clips with the ground truth. Not only was Time-Slicing shown to be fast, the quality of its annotations were very close to that of

the manual approach as discussed in Chap. 4. Finally, we described the large database of video that was put together to help evaluate existing lane detectors. Following the motivation to establish the database, we specified the requirements of a complete database. Then, we took a close look at the existing databases and discussed their provided features. Subsequently, we provided various details of our database such as the hardware acquisition platform used to record video, and the post processing steps that were used to trim and annotate the recorded content. Next, we discussed the available content and provided access information the database. Finally, we summarized the contributions of the publications that formulated the basis of the research in this dissertation.

7.2 Contributions of the Thesis

The contributions of the dissertation are summarized below:

- Developed three new lane detection methodologies.
- Presented the Polar Randomized Hough Transform (PRHT) that extends the Randomized Hough Transform (RHT) by allowing a line to be described directly using polar co-ordinates without an intermediate slope-intercept representation.
- Developed a loose constraint based parallel line detector.
- Presented an error calculation method that enables objective evaluations of the lane detectors.
- Developed a new method to calibrate non-overlapping camera networks using static markers on the road.
- Presented the Multi-Camera Lane Departure Warning (MCLDW) system. The MCLDW system accurately estimates the position of the vehicle on the road with respect to the lane boundaries.

- Devised spline replication process to estimate the lane boundary in the case of missing data.
- Devised boundary extension process to estimate the lane boundary in the case of missing data.
- Developed the Time-Slicing technique that greatly cuts down the time spent in annotating the ground truth while also maintaining quality that is very close to the manual approach.
- Created a very large database of video clips to help standardize data sets that are used in training and testing a variety of lane detectors.
- Provided ground truth for each video clip in XML format.

7.3 Avenues for Future Work

Although we have provided good solutions for the MCLDW system and the evaluation framework, there are numerous avenues to improve the presented research. Some of these avenues are discussed in following subsections.

7.3.1 Lane Detection

Once the calibration parameters of the camera have been determined, they are left untouched. However, while driving on highways and city roads, the vehicle encounters vibrations which could temporarily modify these parameters. As a result, some type of self or auto calibration procedure could be useful. This could be performed when it is determined that the steering wheel of the vehicle is dead centered and the vehicle is commuting at a reasonably high speed. For example, the IPM equations could be modified to solve for height, tilt, and yaw angles such that the lane boundaries appear parallel. Furthermore, we could set a constraint such that the lane boundaries appear to be separated by specified distance.

Another area for improvement is the Temporal Blurring. As mentioned in Sec. 2.5, the parameters of the Temporal Blurring are fixed which does not always yield the best results. Once additional information such as GPS co-ordinates and steering angle are made available, ego motion of the vehicle could be determined. Moreover, the ego motion could used to make the Temporal Blurring adaptive by applying dynamics to the parameters. For example, rather than using a fixed number of images to create the Average Image, the blurring algorithm could adjust the number of images being used. Additionally, a spatial shift could be applied to each image to account for lateral movement performed by the vehicle. Together, both of these enhancements could help detect both solid and dashed lane markers.

The Polar Randomized Hough Transform (PRHT) presented in Sec. 2.6.4 is an extension to the Randomized Hough Transform (RHT) that avoids the need to represent a line using the slope intercept form even at an intermediate stage. One of the caveats of this method is that the point pairs are picked at random. As a result, there are times when the point pairs do not belong to the same line leading to an inaccurate result. In the world images, the lane boundaries appear as horizontal lines or otherwise lines that have a slight angle; therefore, the PRHT could be tailored to this particular application by restricting the search space for the point pairs. By windowing the search space such that the point pairs abide to a certain distance requirement, the two points would more likely belong to the same line. This would lead to a fast and more accurate line detector in addition to a less fragmented Hough space.

Lane detection is mostly performed on grayscale or color images that have been acquired in the visible spectrum. The visible spectrum is able provide images on which lane markers can be detected and extracted with good accuracy; however, this spectrum tends to work well when the road surface is dry or mildly damp. In the presence of rain, snow, and inclement weather, most existing lane detectors are unable

to differentiate between lane markers and reflections on the road. These problems could possibly be eliminated by operating in Near-Infrared or thermal spectrums in addition to the visible spectrum. For example, on the road surface, lane markers as well as vehicles on the road are all composed of different materials leading to different levels of heat absorption. As a result, we believe that in thermal images, temperature differences in each object would be noticeable in the different weather conditions. In addition, the other spectrums could help in road detection. For example, to determine where the road surface ends and where the sidewalk or other non-driveable surfaces begin.

Besides relying purely on computer vision, the performance of a lane detector could be further improved with the aid of a smart infrastructure. One example of a smart infrastructure is Road Nails [94, 95] or networked lane markers. These lane markers could have the ability to communicate directly with each vehicle; as a result, the vehicle could determine its position relative to the lane marker regardless of the weather condition providing better localization.

7.3.2 Multi-Camera Applications

In the dissertation, data acquired from the front and the rear cameras is combined to determine the position of the vehicle on the road. Estimating the position can be improved with the help of additional data provided by side facing cameras. The side facing cameras could also be used to provide a 360° view around the vehicle. Besides just providing the driver with a better visualization of the vehicle with respect to the road, this all around view could be used to develop other assistance applications such as blind spot monitoring, lane change assistance, and parking assistance, the last of which has recently entered the consumer market.

7.3.3 Ground Truth

With the introduction of Time-Slicing, the time required to generate ground truth is greatly reduced. In this method, the user only marks the centers of the lane markings. As a result, there is no additional information provided regarding the thickness of the markings. Therefore, the underlying assumption during testing is that lane markers maintain a constant thickness. However, lane markers have a variety thicknesses depending on their purpose based on FHAs specification [19]. As a result, it would be beneficial to provide either the thickness of the lane marker or the locations of the borders of the lane markers in the ground truth. The latter could be provided by extending the current Time-Slicing procedure. When a user clicks to specify control points, a search could be performed to find the locations of the gradients or the edges in the neighborhood of each control point. Assuming that the intensity values within a lane marker are near constant, the edges on either side of the control point would correspond to the borders of the lane marker. By providing border information, the calculation of error as described in Sec. 2.4.8 could be made more accurate without having the need to explicitly specify the thickness of the lane markers.

7.3.4 Database

The motivation for setting up the database is to attempt to standardize the data that is used for training and testing. Currently, video data has only been captured during dry and damp weather conditions. This has restricted testing the lane detectors to these particular conditions. In the future, recording sessions will be performed in the presence of rain, snow, and fog to enable training and testing during inclement weather conditions. By adding hardware to the data acquisition platform, we could record additional sensory information from the vehicle such as GPS co-ordinates and steering angle. This access to additional resources can have a direct impact on existing research as well as research presented in the dissertation. Additionally, the captured

data is currently used purely for vision based lane detection research. We feel that the comprehensive view of the road allows this data to be used for other automotive vision assistance applications e.g. street sign, pedestrian, and vehicle detection to name a few.

In this dissertation, we have presented several single camera and multi-camera solutions for Driver Assistance systems. Although the presented solutions have shown good results, lane detection and LDW are far from being considered as solved problems. The research presented in this document can be significantly enhanced, and some of the avenues for its improvement were discussed above.

APPENDIX A

COMPREHENSIVE TESTING OF ALD 1.0

A histogram of E(n) for each clip is provided below. The standard deviation of the error is less than 1 inch for most clips. This implies that the estimated lane boundaries are minimally misaligned with regard to the ground truth.

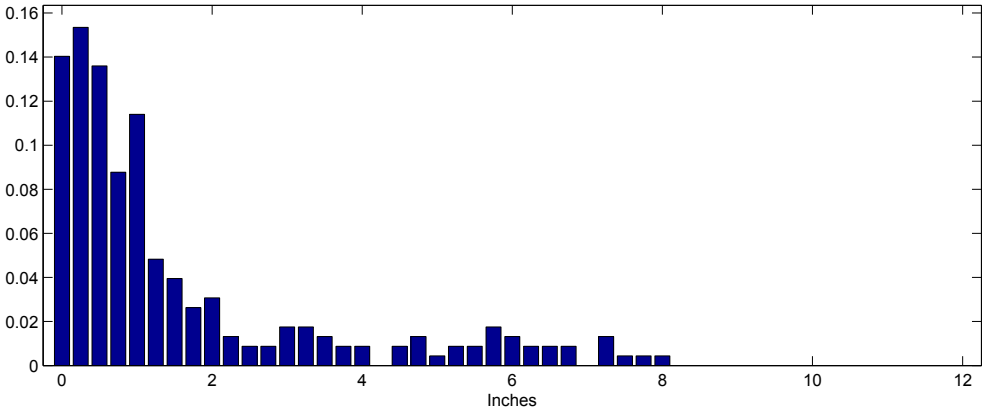


Figure 198: Histogram of E(n) for Clip 1. $\sigma_{\mathrm{E(n)}} = 1.93$

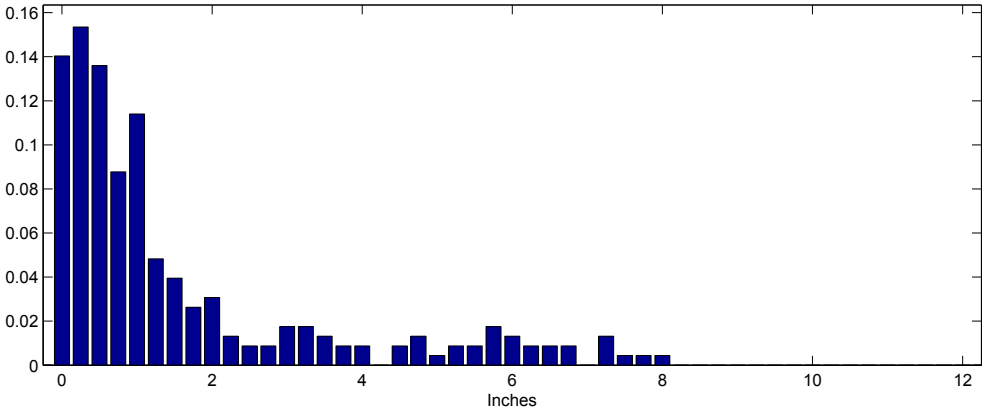


Figure 199: Histogram of E(n) for Clip 2. $\sigma_{\mathrm{E(n)}} = 0.9$

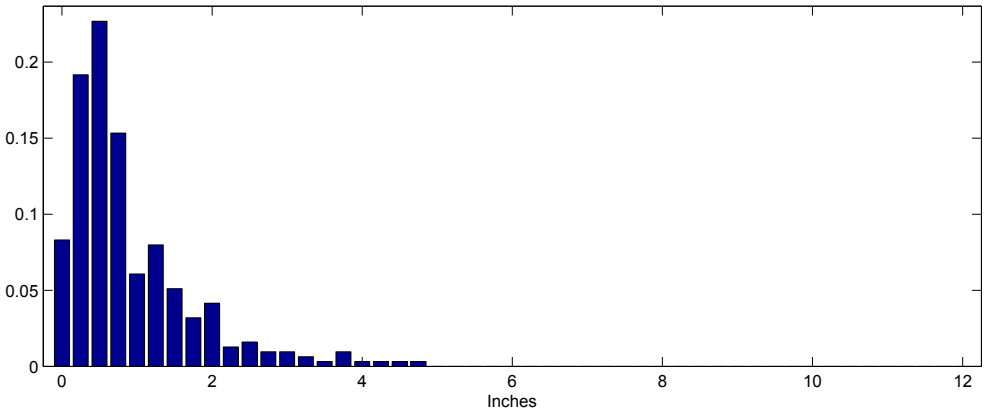


Figure 200: Histogram of E(n) for Clip 3. $\sigma_{\mathrm{E(n)}} = 0.84$

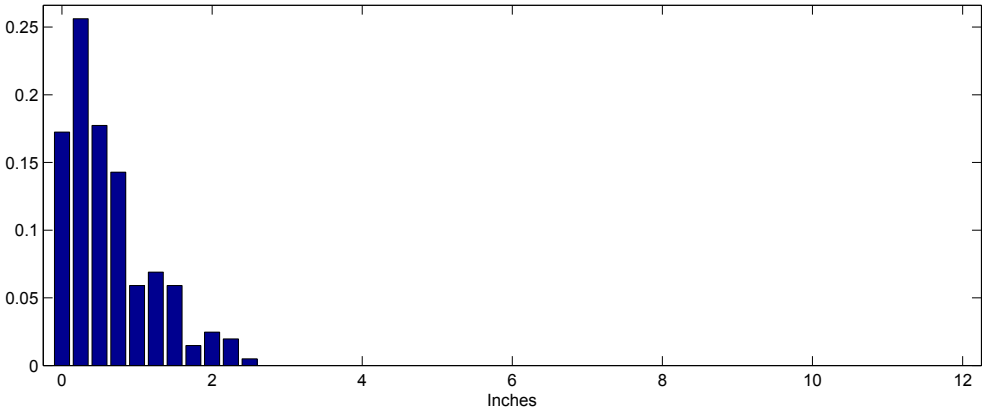


Figure 201: Histogram of E(n) for Clip 4. $\sigma_{\mathrm{E(n)}} = 0.56$

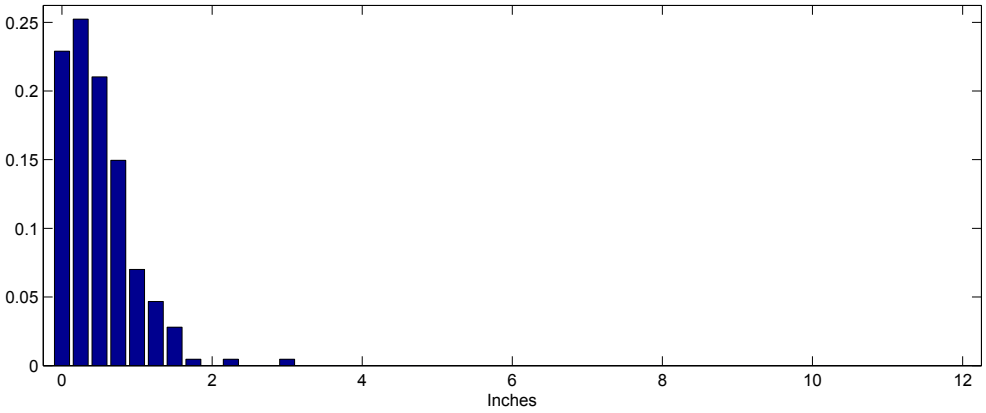


Figure 202: Histogram of E(n) for Clip 5. $\sigma_{\mathrm{E(n)}} = 0.43$

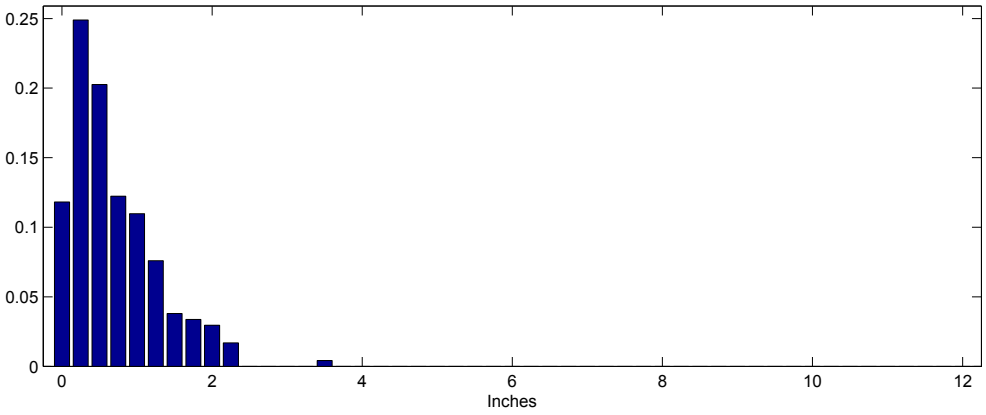


Figure 203: Histogram of E(n) for Clip 6. $\sigma_{\mathrm{E(n)}} = 0.58$

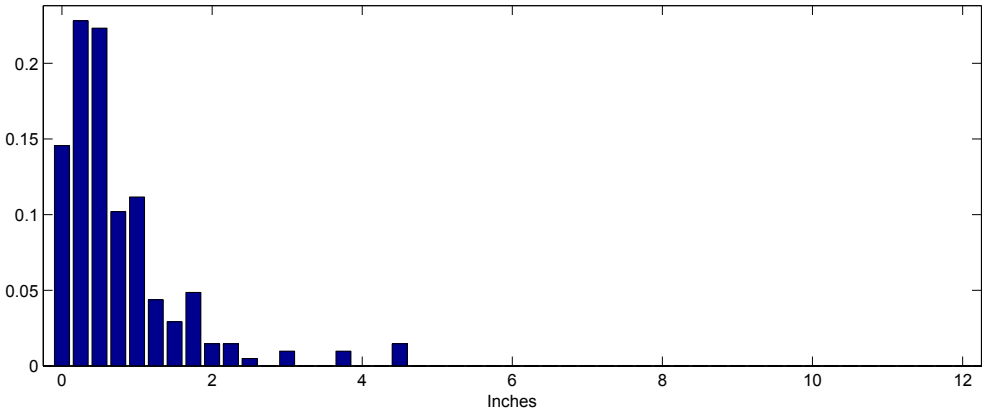


Figure 204: Histogram of E(n) for Clip 7. $\sigma_{\mathrm{E(n)}} = 0.80$

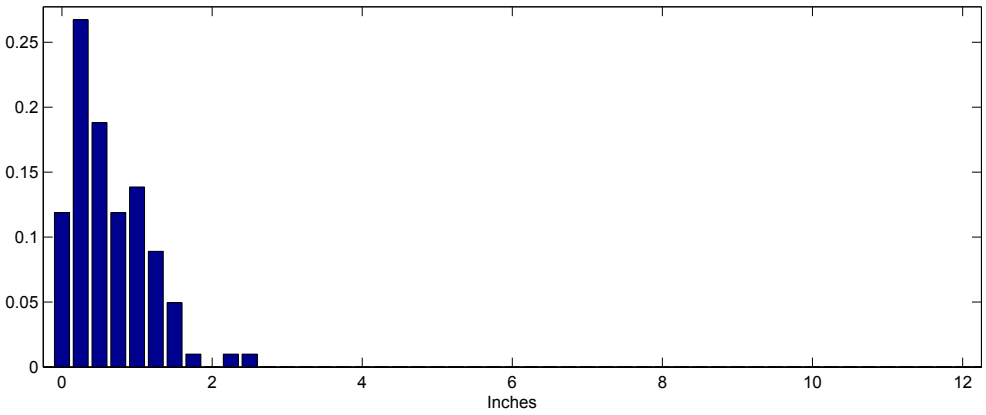


Figure 205: Histogram of E(n) for Clip 8. $\sigma_{\mathrm{E(n)}} = 0.49$

For definitions of the performance measures CO, MA, MD, FD, and TR, please refer to Sec. 2.4.9. Image count is included in the parenthesis.

Table 24: Performance of the ALD 1.0 on Clip 1 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	48.40	50.55	1.04	100.00 (61)	0.00 (0)
$\alpha = 12$, $k = 3$	41.43	45.88	12.69	83.87 (46)	16.13 (15)
$\alpha = 17$, $k = 3$	11.83	29.25	58.93	10.22 (10)	89.78 (52)
$\alpha = 22$, $k = 3$	0.69	6.60	92.71	0.00 (0)	100.00 (61)
$\alpha = 27$, $k = 3$	0.00	0.00	100.00	0.00 (0)	100.00 (61)
$\alpha = 7$, $k = 5$	47.43	51.53	1.04	100.00 (61)	0.00 (0)
$\alpha = 12$, $k = 5$	47.54	51.41	1.04	100.00 (61)	0.00 (0)
$\alpha = 17$, $k = 5$	42.95	56.01	1.04	100.00 (61)	0.00 (0)
$\alpha = 22$, $k = 5$	34.21	45.75	20.04	82.80 (45)	17.20 (16)
$\alpha = 27$, $k = 5$	19.35	38.26	42.39	17.43 (14)	82.57 (47)
$\alpha = 7$, $k = 7$	47.49	51.46	1.04	100.00 (61)	0.00 (0)
$\alpha = 12$, $k = 7$	48.42	50.54	1.04	100.00 (61)	0.00 (0)
$\alpha = 17$, $k = 7$	45.98	52.98	1.04	100.00 (61)	0.00 (0)
$\alpha = 22$, $k = 7$	42.40	55.32	2.28	100.00 (61)	0.00 (0)
$\alpha = 27$, $k = 7$	32.10	47.03	20.88	82.80 (45)	17.20 (16)
$\alpha = 7$, $k = 9$	48.26	50.70	1.04	100.00 (61)	0.00 (0)
$\alpha = 12$, $k = 9$	48.75	50.21	1.04	100.00 (61)	0.00 (0)
$\alpha = 17$, $k = 9$	47.22	51.73	1.04	100.00 (61)	0.00 (0)
$\alpha = 22$, $k = 9$	42.39	55.76	1.85	100.00 (61)	0.00 (0)
$\alpha = 27$, $k = 9$	34.12	48.13	17.75	83.87 (46)	16.13 (15)
$\alpha = 7$, $k = 11$	48.87	50.08	1.04	100.00 (61)	0.00 (0)
$\alpha = 12$, $k = 11$	47.51	51.45	1.04	100.00 (61)	0.00 (0)
$\alpha = 17$, $k = 11$	45.35	53.61	1.04	100.00 (61)	0.00 (0)
$\alpha = 22$, $k = 11$	42.20	56.08	1.73	100.00 (61)	0.00 (0)
$\alpha = 27$, $k = 11$	36.49	46.56	16.95	90.32 (52)	9.68 (9)

Table 25: Performance of the ALD 1.0 on Clip 2 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	81.10	17.62	1.28	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 3$	66.05	32.67	1.28	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 3$	33.38	58.99	7.63	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 3$	8.42	66.55	25.04	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 3$	0.00	48.22	51.78	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 5$	83.38	15.34	1.28	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 5$	81.81	16.90	1.28	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 5$	73.75	24.96	1.28	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 5$	61.41	37.30	1.28	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 5$	39.16	56.49	4.35	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 7$	83.52	15.19	1.28	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 7$	82.03	16.69	1.28	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 7$	78.32	20.40	1.28	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 7$	69.76	28.96	1.28	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 7$	53.50	45.22	1.28	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 9$	84.95	13.77	1.28	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 9$	81.95	16.76	1.28	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 9$	78.67	20.04	1.28	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 9$	72.90	25.82	1.28	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 9$	56.92	41.80	1.28	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 11$	84.38	14.34	1.28	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 11$	81.74	16.98	1.28	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 11$	79.32	19.40	1.28	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 11$	73.25	25.46	1.28	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 11$	57.70	41.01	1.28	0.00 (0)	0.00 (0)

Table 26: Performance of the ALD 1.0 on Clip 3 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	83.50	15.13	1.37	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 3$	61.54	33.97	4.50	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 3$	28.59	52.70	18.72	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 3$	11.66	53.29	35.05	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 3$	2.96	55.39	41.65	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 5$	88.21	10.42	1.37	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 5$	84.42	14.21	1.37	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 5$	72.56	25.53	1.90	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 5$	54.31	36.39	9.31	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 5$	34.83	46.68	18.49	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 7$	83.57	14.98	1.45	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 7$	79.78	17.10	3.13	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 7$	79.94	18.69	1.37	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 7$	66.48	28.26	5.26	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 7$	51.72	38.07	10.22	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 9$	74.51	24.04	1.45	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 9$	78.18	19.61	2.21	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 9$	72.33	23.55	4.12	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 9$	69.90	25.15	4.96	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 9$	53.69	36.39	9.91	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 11$	67.13	31.50	1.37	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 11$	74.61	23.41	1.98	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 11$	70.97	24.84	4.19	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 11$	62.52	31.30	6.18	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 11$	53.92	37.15	8.93	0.00 (0)	0.00 (0)

Table 27: Performance of the ALD 1.0 on Clip 4 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	67.28	27.54	5.18	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 3$	52.96	25.69	21.35	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 3$	30.68	16.45	52.87	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 3$	19.69	14.97	65.34	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 3$	8.78	23.57	67.65	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 5$	76.62	21.72	1.66	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 5$	75.05	19.69	5.27	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 5$	63.31	17.93	18.76	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 5$	44.73	32.07	23.20	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 5$	29.11	20.52	50.37	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 7$	73.84	24.49	1.66	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 7$	70.79	26.62	2.59	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 7$	65.71	17.65	16.64	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 7$	55.36	24.21	20.43	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 7$	38.26	26.25	35.49	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 9$	73.75	24.58	1.66	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 9$	64.70	33.64	1.66	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 9$	64.05	18.67	17.28	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 9$	58.60	24.21	17.19	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 9$	44.92	25.23	29.85	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 11$	73.75	24.58	1.66	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 11$	62.01	36.32	1.66	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 11$	59.33	25.05	15.62	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 11$	60.91	21.63	17.47	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 11$	42.61	36.97	20.43	0.00 (0)	0.00 (0)

Table 28: Performance of the ALD 1.0 on Clip 5 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	91.32	5.47	3.21	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 3$	88.24	7.42	4.34	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 3$	56.54	19.62	23.84	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 3$	39.18	6.10	54.72	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 3$	14.53	16.35	69.12	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 5$	90.19	7.17	2.64	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 5$	91.57	5.22	3.21	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 5$	89.94	6.73	3.33	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 5$	84.03	11.38	4.59	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 5$	58.49	21.19	20.31	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 7$	90.50	6.86	2.64	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 7$	91.07	6.29	2.64	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 7$	91.01	5.79	3.21	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 7$	88.49	7.99	3.52	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 7$	76.48	16.29	7.23	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 9$	90.50	6.86	2.64	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 9$	90.69	6.67	2.64	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 9$	91.64	5.16	3.21	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 9$	90.13	6.60	3.27	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 9$	78.36	14.47	7.17	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 11$	89.43	7.92	2.64	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 11$	89.18	8.18	2.64	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 11$	89.12	7.67	3.21	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 11$	90.88	5.85	3.27	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 11$	81.26	14.09	4.65	0.00 (0)	0.00 (0)

Table 29: Performance of the ALD 1.0 on Clip 6 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	79.59	18.37	2.04	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 3$	71.09	26.87	2.04	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 3$	66.89	18.59	14.51	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 3$	41.04	41.84	17.12	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 3$	16.33	35.26	48.41	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 5$	82.65	15.31	2.04	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 5$	80.05	17.91	2.04	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 5$	81.52	16.44	2.04	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 5$	69.16	17.23	13.61	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 5$	64.63	19.61	15.76	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 7$	82.77	15.19	2.04	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 7$	83.67	14.29	2.04	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 7$	74.94	23.02	2.04	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 7$	73.47	24.49	2.04	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 7$	68.48	19.95	11.56	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 9$	81.63	16.33	2.04	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 9$	81.07	16.89	2.04	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 9$	76.08	21.88	2.04	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 9$	70.86	27.10	2.04	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 9$	67.01	30.95	2.04	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 11$	78.00	19.95	2.04	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 11$	81.86	16.10	2.04	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 11$	74.94	23.02	2.04	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 11$	72.34	25.62	2.04	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 11$	63.72	34.24	2.04	0.00 (0)	0.00 (0)

Table 30: Performance of the ALD 1.0 on Clip 7 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 3$	97.44	0.97	1.59	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 3$	96.38	2.03	1.59	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 3$	94.36	4.06	1.59	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 3$	80.16	17.55	2.29	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 5$	96.74	1.68	1.59	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 5$	97.44	0.97	1.59	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 5$	97.71	0.71	1.59	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 5$	97.35	1.06	1.59	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 5$	97.44	0.97	1.59	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 7$	96.65	1.76	1.59	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 7$	96.47	1.94	1.59	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 7$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 7$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 7$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 9$	96.47	1.94	1.59	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 9$	97.09	1.32	1.59	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 9$	97.27	1.15	1.59	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 9$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 9$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 7$, $k = 11$	96.91	1.50	1.59	0.00 (0)	0.00 (0)
$\alpha = 12$, $k = 11$	97.18	1.23	1.59	0.00 (0)	0.00 (0)
$\alpha = 17$, $k = 11$	97.62	0.79	1.59	0.00 (0)	0.00 (0)
$\alpha = 22$, $k = 11$	97.53	0.88	1.59	0.00 (0)	0.00 (0)
$\alpha = 27$, $k = 11$	97.35	1.06	1.59	0.00 (0)	0.00 (0)

Table 31: Performance of the ALD 1.0 on Clip 8 for different values of α and k.

Parameters	CO %	MA %	MD %	FD %	TR %
$\alpha = 7$, $k = 3$	85.48	13.10	1.42	100.00 (65)	0.00 (0)
$\alpha = 12$, $k = 3$	94.67	3.91	1.42	100.00 (65)	0.00 (0)
$\alpha = 17$, $k = 3$	96.77	1.81	1.42	100.00 (65)	0.00 (0)
$\alpha = 22$, $k = 3$	96.61	1.97	1.42	100.00 (65)	0.00 (0)
$\alpha = 27$, $k = 3$	91.45	7.13	1.42	100.00 (65)	0.00 (0)
$\alpha = 7$, $k = 5$	71.00	27.59	1.42	100.00 (65)	0.00 (0)
$\alpha = 12$, $k = 5$	83.61	14.97	1.42	100.00 (65)	0.00 (0)
$\alpha = 17$, $k = 5$	92.18	6.40	1.42	100.00 (65)	0.00 (0)
$\alpha = 22$, $k = 5$	94.44	4.15	1.42	100.00 (65)	0.00 (0)
$\alpha = 27$, $k = 5$	95.45	3.13	1.42	100.00 (65)	0.00 (0)
$\alpha = 7$, $k = 7$	69.67	28.91	1.42	100.00 (65)	0.00 (0)
$\alpha = 12$, $k = 7$	72.24	26.34	1.42	100.00 (65)	0.00 (0)
$\alpha = 17$, $k = 7$	83.61	14.97	1.42	100.00 (65)	0.00 (0)
$\alpha = 22$, $k = 7$	91.79	6.79	1.42	100.00 (65)	0.00 (0)
$\alpha = 27$, $k = 7$	93.97	4.61	1.42	100.00 (65)	0.00 (0)
$\alpha = 7$, $k = 9$	68.74	29.85	1.42	100.00 (65)	0.00 (0)
$\alpha = 12$, $k = 9$	69.83	28.75	1.42	100.00 (65)	0.00 (0)
$\alpha = 17$, $k = 9$	75.44	23.15	1.42	100.00 (65)	0.00 (0)
$\alpha = 22$, $k = 9$	86.18	12.40	1.42	100.00 (65)	0.00 (0)
$\alpha = 27$, $k = 9$	91.01	7.57	1.42	100.00 (65)	0.00 (0)
$\alpha = 7$, $k = 11$	68.97	29.61	1.42	100.00 (65)	0.00 (0)
$\alpha = 12$, $k = 11$	69.67	28.91	1.42	100.00 (65)	0.00 (0)
$\alpha = 17$, $k = 11$	72.25	26.34	1.42	100.00 (65)	0.00 (0)
$\alpha = 22$, $k = 11$	80.89	17.69	1.42	100.00 (65)	0.00 (0)
$\alpha = 27$, $k = 11$	89.22	9.37	1.42	100.00 (65)	0.00 (0)

APPENDIX B

COMPREHENSIVE TESTING OF ALD 2.0

A histogram of E(n) for each clip is provided below. The standard deviation of the error is less than 1.5 inches for most clips. This implies that the estimated lane boundaries are minimally misaligned with regard to the ground truth.

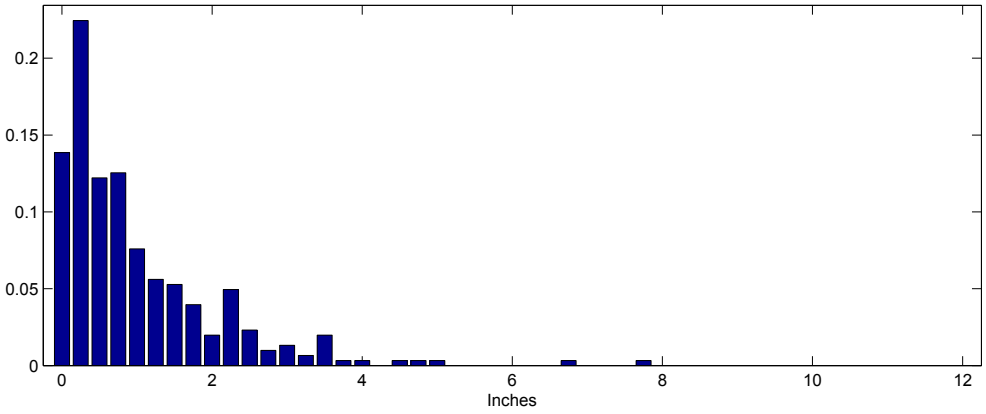


Figure 206: Histogram of E(n) for Clip 1. $\sigma_{\mathrm{E(n)}} = 1.08$

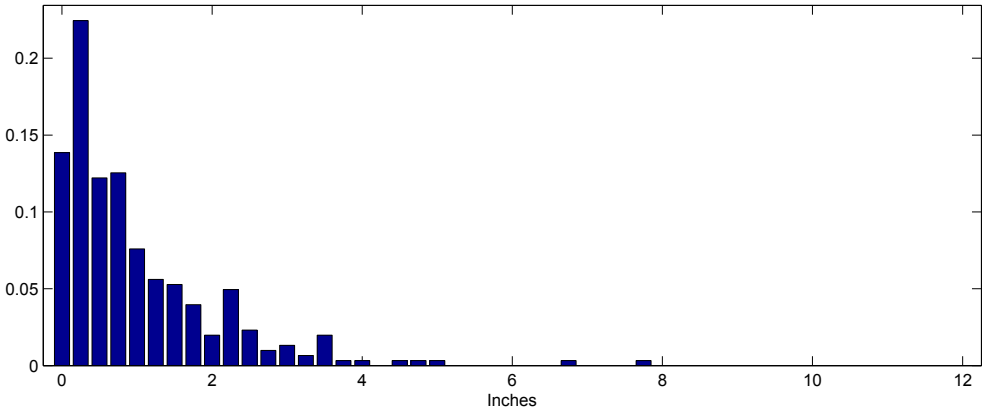


Figure 207: Histogram of E(n) for Clip 2. $\sigma_{\mathrm{E(n)}} = 2.14$

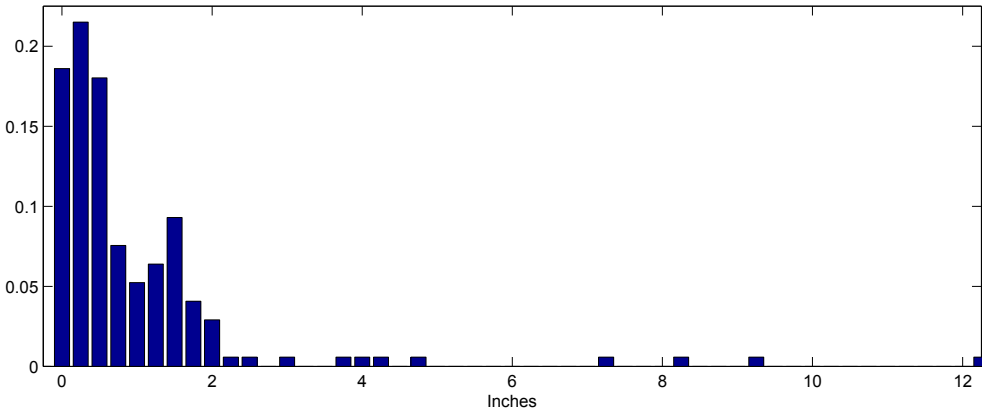


Figure 208: Histogram of E(n) for Clip 3. $\sigma_{\mathrm{E(n)}} = 1.63$

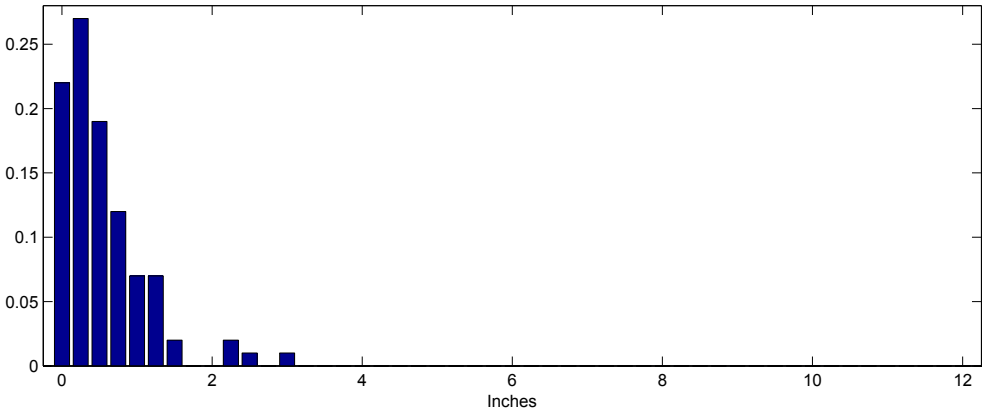


Figure 209: Histogram of E(n) for Clip 4. $\sigma_{\mathrm{E(n)}} = 0.56$

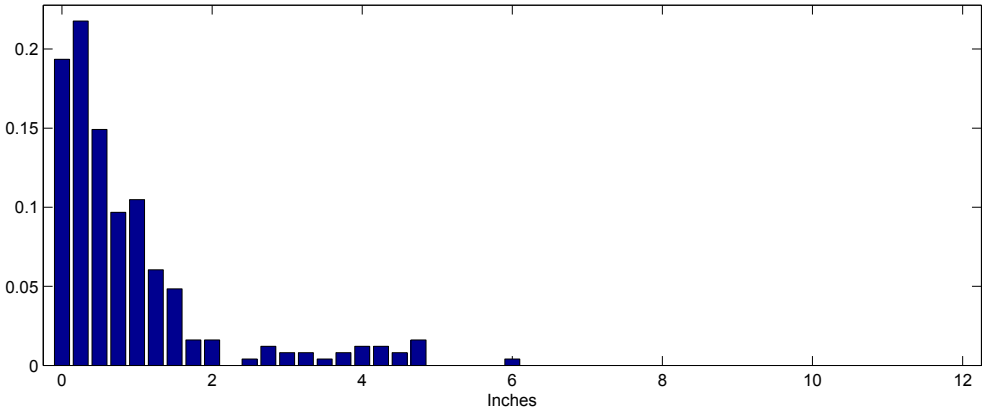


Figure 210: Histogram of E(n) for Clip 5. $\sigma_{\mathrm{E(n)}} = 1.11$

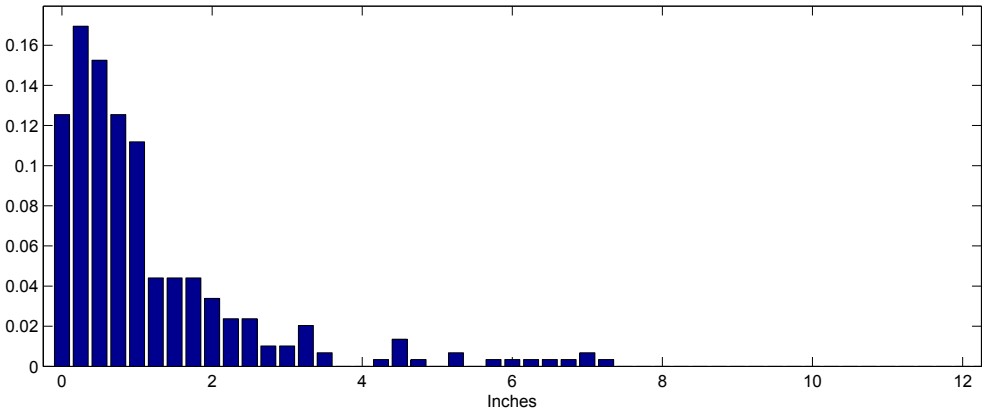


Figure 211: Histogram of E(n) for Clip 6. $\sigma_{\mathrm{E(n)}} = 1.35$

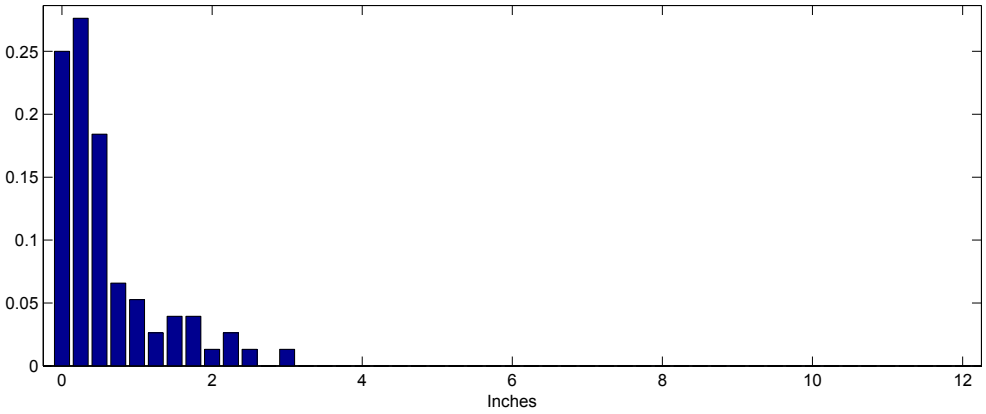


Figure 212: Histogram of E(n) for Clip 7. $\sigma_{\mathrm{E(n)}} = 0.67$

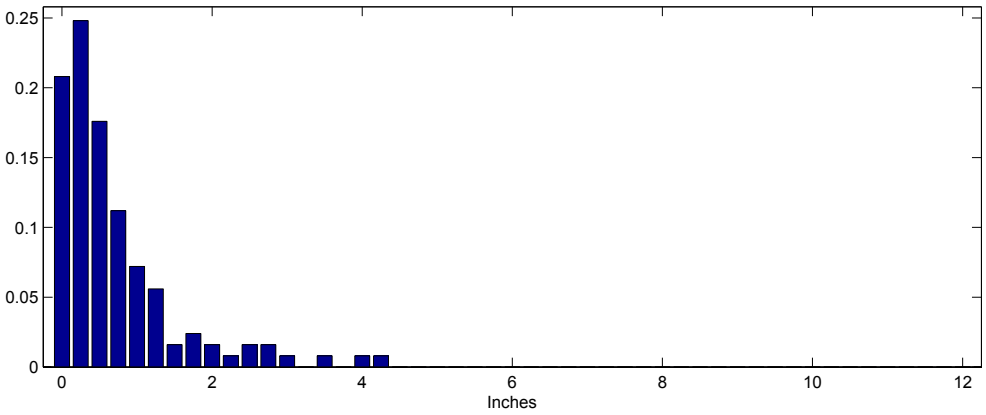


Figure 213: Histogram of E(n) for Clip 8. $\sigma_{\mathrm{E(n)}} = 0.81$

For definitions of the performance measures CO, MA, MD, FD, and TR, please refer to Sec. 2.4.9. Image count is included in the parenthesis.

Table 32: Performance of the ALD 2.0 on Clip 1 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	89.97	9.60	0.43	100.00 (61)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	89.97	9.60	0.43	100.00 (61)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	89.97	9.60	0.43	100.00 (61)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	89.97	9.60	0.43	100.00 (61)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	89.97	9.60	0.43	100.00 (61)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	89.44	10.13	0.43	100.00 (61)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	89.44	10.13	0.43	100.00 (61)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	89.44	10.13	0.43	100.00 (61)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	89.44	10.13	0.43	100.00 (61)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	89.44	10.13	0.43	100.00 (61)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	89.50	10.07	0.43	100.00 (61)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	89.50	10.07	0.43	100.00 (61)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	89.50	10.07	0.43	100.00 (61)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	89.50	10.07	0.43	100.00 (61)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	89.50	10.07	0.43	100.00 (61)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	89.39	10.18	0.43	100.00 (61)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	89.39	10.18	0.43	100.00 (61)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	89.39	10.18	0.43	100.00 (61)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	89.39	10.18	0.43	100.00 (61)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	89.39	10.18	0.43	100.00 (61)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	90.37	9.20	0.43	100.00 (61)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	90.37	9.20	0.43	100.00 (61)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	90.37	9.20	0.43	100.00 (61)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	90.37	9.20	0.43	100.00 (61)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	90.37	9.20	0.43	100.00 (61)	0.00 (0)

Table 33: Performance of the ALD 2.0 on Clip 2 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	88.80	10.98	0.21	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	88.80	10.98	0.21	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	88.80	10.98	0.21	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	88.80	10.98	0.21	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	88.80	10.98	0.21	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	89.16	10.49	0.36	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	89.16	10.49	0.36	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	89.16	10.49	0.36	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	89.16	10.49	0.36	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	89.16	10.49	0.36	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	88.52	11.13	0.36	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	88.52	11.13	0.36	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	88.52	11.13	0.36	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	88.52	11.13	0.36	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	88.52	11.13	0.36	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	88.80	10.77	0.43	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	88.80	10.77	0.43	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	88.80	10.77	0.43	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	88.80	10.77	0.43	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	88.80	10.77	0.43	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	88.94	10.63	0.43	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	88.94	10.63	0.43	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	88.94	10.63	0.43	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	88.94	10.63	0.43	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	88.94	10.63	0.43	0.00 (0)	0.00 (0)

Table 34: Performance of the ALD 2.0 on Clip 3 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	86.30	13.09	0.61	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	86.22	13.17	0.61	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	86.37	13.02	0.61	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	86.30	13.09	0.61	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	86.30	13.09	0.61	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	87.44	11.95	0.61	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	87.51	11.88	0.61	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	87.44	11.95	0.61	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	87.51	11.88	0.61	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	87.51	11.88	0.61	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	88.96	10.43	0.61	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	88.96	10.43	0.61	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	89.03	10.36	0.61	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	89.03	10.36	0.61	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	88.88	10.51	0.61	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	90.70	8.61	0.69	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	90.55	8.76	0.69	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	90.48	8.84	0.69	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	90.48	8.84	0.69	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	90.40	8.91	0.69	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	90.55	8.76	0.69	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	90.70	8.61	0.69	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	90.70	8.61	0.69	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	90.55	8.76	0.69	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	90.63	8.68	0.69	0.00 (0)	0.00 (0)

Table 35: Performance of the ALD 2.0 on Clip 4 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	95.56	4.34	0.09	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	95.56	4.34	0.09	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	95.56	4.34	0.09	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	95.56	4.34	0.09	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	95.56	4.34	0.09	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	96.67	3.23	0.09	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	96.67	3.23	0.09	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	96.67	3.23	0.09	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	96.67	3.23	0.09	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	96.67	3.23	0.09	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	97.23	2.68	0.09	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	97.23	2.68	0.09	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	97.23	2.68	0.09	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	97.23	2.68	0.09	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	97.23	2.68	0.09	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	97.41	2.50	0.09	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	97.41	2.50	0.09	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	97.41	2.50	0.09	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	97.41	2.50	0.09	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	97.41	2.50	0.09	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	97.97	1.94	0.09	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	97.97	1.94	0.09	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	97.97	1.94	0.09	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	97.97	1.94	0.09	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	97.97	1.94	0.09	0.00 (0)	0.00 (0)

Table 36: Performance of the ALD 2.0 on Clip 5 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	96.54	3.27	0.19	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	96.54	3.27	0.19	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	96.54	3.27	0.19	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	96.54	3.27	0.19	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	96.54	3.27	0.19	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	96.23	3.58	0.19	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	96.23	3.58	0.19	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	96.23	3.58	0.19	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	96.23	3.58	0.19	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	96.23	3.58	0.19	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	96.10	3.71	0.19	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	96.10	3.71	0.19	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	96.10	3.71	0.19	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	96.10	3.71	0.19	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	96.10	3.71	0.19	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	95.41	4.40	0.19	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	95.41	4.40	0.19	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	95.41	4.40	0.19	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	95.41	4.40	0.19	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	95.41	4.40	0.19	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	94.72	5.09	0.19	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	94.72	5.09	0.19	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	94.72	5.09	0.19	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	94.72	5.09	0.19	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	94.72	5.09	0.19	0.00 (0)	0.00 (0)

Table 37: Performance of the ALD 2.0 on Clip 6 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	86.96	12.70	0.34	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	86.96	12.70	0.34	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	86.96	12.70	0.34	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	86.96	12.70	0.34	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	86.96	12.70	0.34	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	89.00	10.66	0.34	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	88.89	10.77	0.34	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	88.89	10.77	0.34	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	89.00	10.66	0.34	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	88.89	10.77	0.34	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	88.55	11.11	0.34	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	88.66	11.00	0.34	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	88.66	11.00	0.34	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	88.66	11.00	0.34	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	88.55	11.11	0.34	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	90.25	9.41	0.34	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	89.91	9.75	0.34	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	89.91	9.75	0.34	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	89.91	9.75	0.34	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	89.91	9.75	0.34	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	89.91	9.75	0.34	0.00 (0)	0.00 (0)

Table 38: Performance of the ALD 2.0 on Clip 7 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	98.85	1.15	0.00	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	98.85	1.15	0.00	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	98.85	1.15	0.00	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	98.85	1.15	0.00	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	98.85	1.15	0.00	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	99.03	0.97	0.00	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	99.29	0.71	0.00	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	99.29	0.71	0.00	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	99.29	0.71	0.00	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	99.29	0.71	0.00	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	99.29	0.71	0.00	0.00 (0)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	99.56	0.44	0.00	0.00 (0)	0.00 (0)

Table 39: Performance of the ALD 2.0 on Clip 8 for different values of δ and ϵ.

Parameters	CO %	MA %	MD %	FD %	TR %
$\delta = 10$, $\epsilon = 0.55$	69.79	30.21	0.00	100.00 (65)	0.00 (0)
$\delta = 20$, $\epsilon = 0.55$	69.79	30.21	0.00	100.00 (65)	0.00 (0)
$\delta = 30$, $\epsilon = 0.55$	69.55	30.45	0.00	100.00 (65)	0.00 (0)
$\delta = 40$, $\epsilon = 0.55$	69.63	30.37	0.00	100.00 (65)	0.00 (0)
$\delta = 50$, $\epsilon = 0.55$	69.55	30.45	0.00	100.00 (65)	0.00 (0)
$\delta = 10$, $\epsilon = 0.65$	73.54	26.46	0.00	100.00 (65)	0.00 (0)
$\delta = 20$, $\epsilon = 0.65$	73.54	26.46	0.00	100.00 (65)	0.00 (0)
$\delta = 30$, $\epsilon = 0.65$	73.22	26.78	0.00	100.00 (65)	0.00 (0)
$\delta = 40$, $\epsilon = 0.65$	73.22	26.78	0.00	100.00 (65)	0.00 (0)
$\delta = 50$, $\epsilon = 0.65$	73.30	26.70	0.00	100.00 (65)	0.00 (0)
$\delta = 10$, $\epsilon = 0.75$	81.80	18.20	0.00	100.00 (65)	0.00 (0)
$\delta = 20$, $\epsilon = 0.75$	81.88	18.12	0.00	100.00 (65)	0.00 (0)
$\delta = 30$, $\epsilon = 0.75$	81.57	18.43	0.00	100.00 (65)	0.00 (0)
$\delta = 40$, $\epsilon = 0.75$	81.57	18.43	0.00	100.00 (65)	0.00 (0)
$\delta = 50$, $\epsilon = 0.75$	81.57	18.43	0.00	100.00 (65)	0.00 (0)
$\delta = 10$, $\epsilon = 0.85$	88.81	11.19	0.00	100.00 (65)	0.00 (0)
$\delta = 20$, $\epsilon = 0.85$	88.89	11.11	0.00	100.00 (65)	0.00 (0)
$\delta = 30$, $\epsilon = 0.85$	88.57	11.43	0.00	100.00 (65)	0.00 (0)
$\delta = 40$, $\epsilon = 0.85$	88.42	11.58	0.00	100.00 (65)	0.00 (0)
$\delta = 50$, $\epsilon = 0.85$	88.34	11.66	0.00	100.00 (65)	0.00 (0)
$\delta = 10$, $\epsilon = 0.95$	91.14	8.86	0.00	100.00 (65)	0.00 (0)
$\delta = 20$, $\epsilon = 0.95$	91.07	8.93	0.00	100.00 (65)	0.00 (0)
$\delta = 30$, $\epsilon = 0.95$	91.14	8.86	0.00	100.00 (65)	0.00 (0)
$\delta = 40$, $\epsilon = 0.95$	91.22	8.78	0.00	100.00 (65)	0.00 (0)
$\delta = 50$, $\epsilon = 0.95$	91.07	8.93	0.00	100.00 (65)	0.00 (0)

APPENDIX C

COMPREHENSIVE TESTING OF ALD 3.0

A histogram of E(n) for each clip is provided below. The standard deviation of the error is less than 1.5 inches for most clips. This implies that the estimated lane boundaries are minimally misaligned with regard to the ground truth.

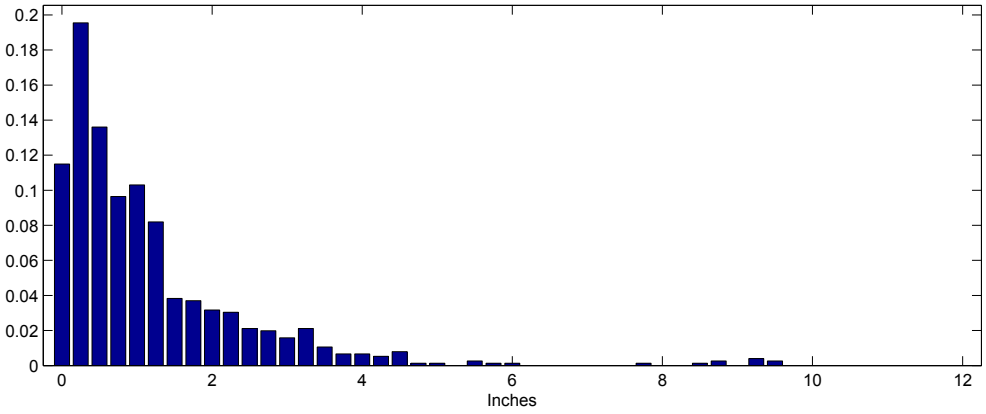


Figure 214: Histogram of E(n) for Clip 1. $\sigma_{\mathrm{E(n)}} = 1.36$

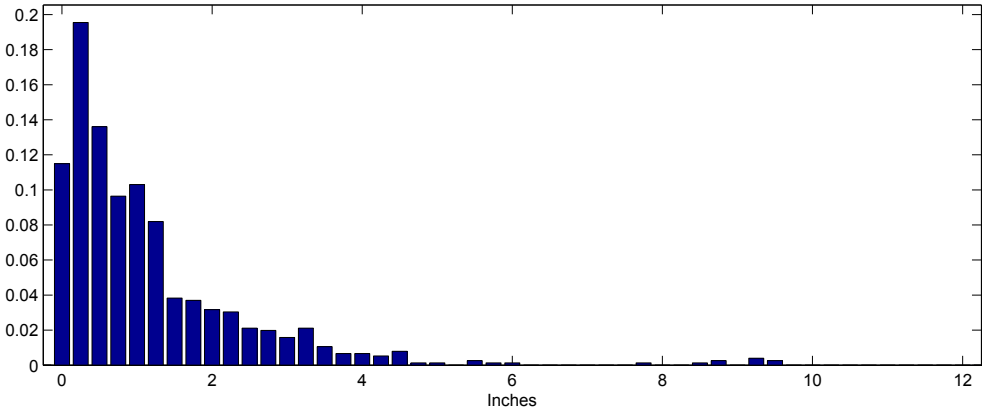


Figure 215: Histogram of E(n) for Clip 2. $\sigma_{\mathrm{E(n)}} = 2.23$

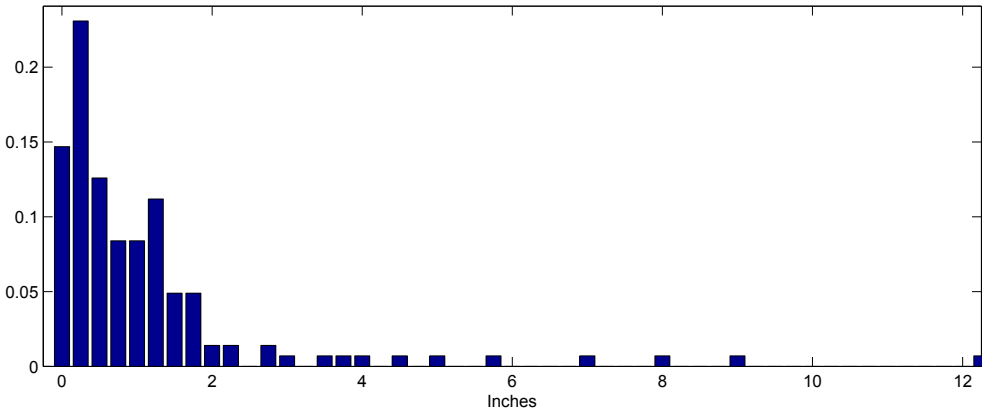


Figure 216: Histogram of E(n) for Clip 3. $\sigma_{\mathrm{E(n)}} = 1.79$

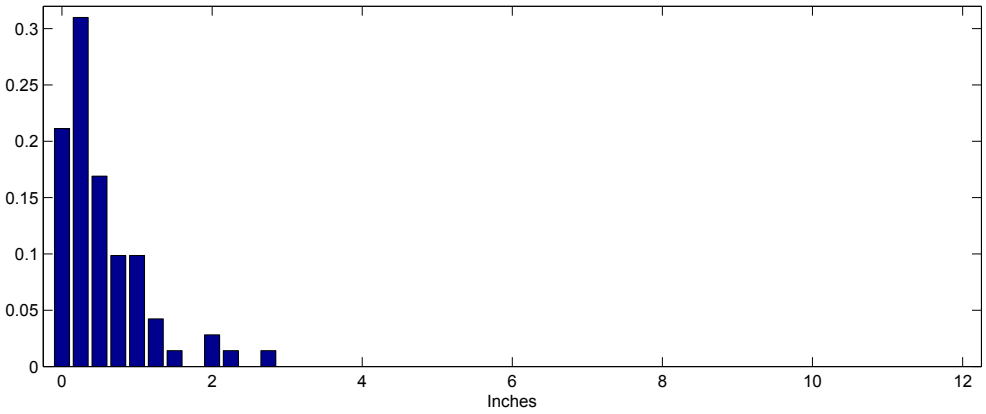


Figure 217: Histogram of E(n) for Clip 4. $\sigma_{\mathrm{E(n)}} = 0.57$

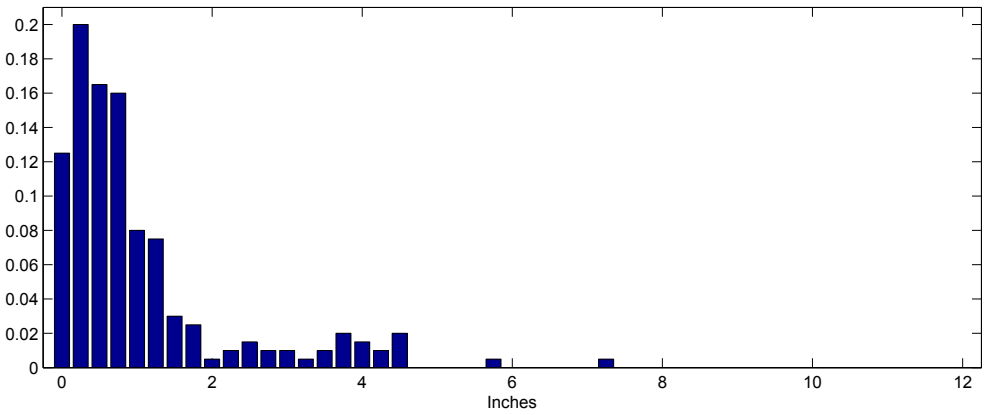


Figure 218: Histogram of E(n) for Clip 5. $\sigma_{\mathrm{E(n)}} = 1.20$

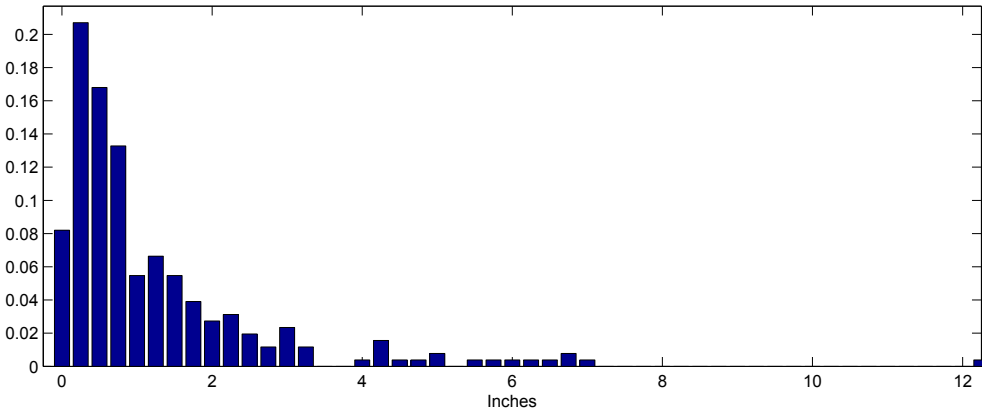


Figure 219: Histogram of E(n) for Clip 6. $\sigma_{\mathrm{E(n)}} = 1.71$

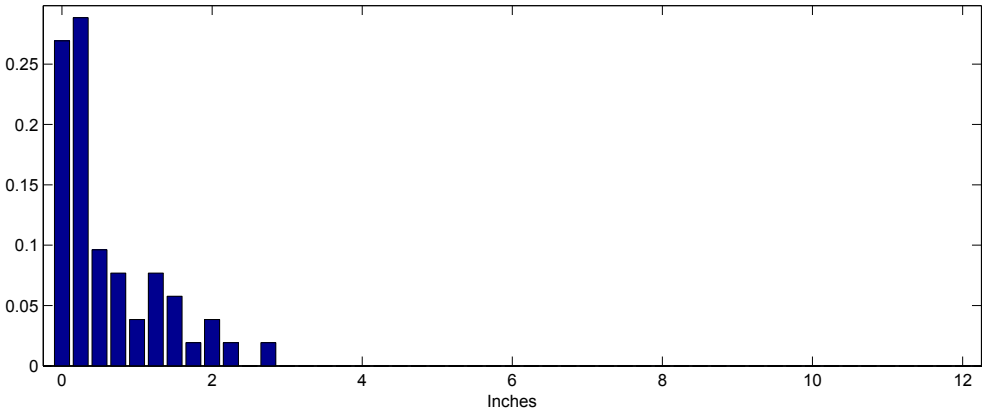


Figure 220: Histogram of E(n) for Clip 7. $\sigma_{\mathrm{E(n)}} = 0.69$

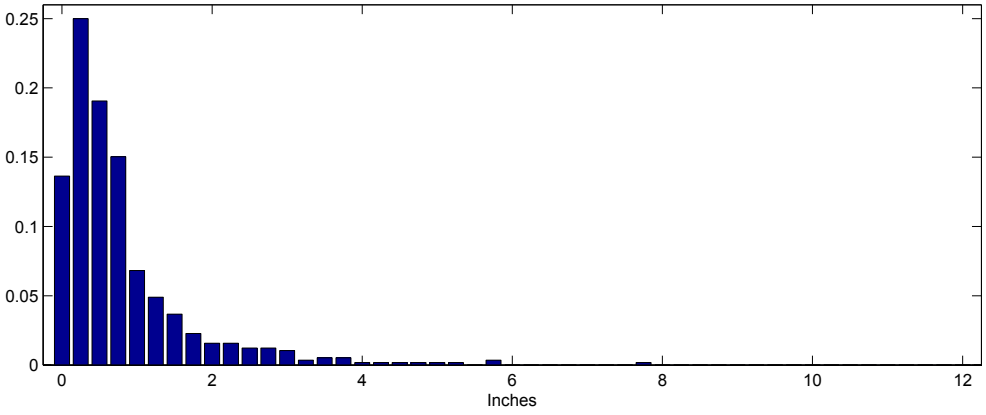


Figure 221: Histogram of E(n) for Clip 8. $\sigma_{\mathrm{E(n)}} = 0.92$

For definitions of the performance measures CO, MA, MD, FD, and TR, please refer to Sec. 2.4.9. Image count is included in the parenthesis.

Table 40: Performance of the ALD 3.0 on Clip 1 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	87.42	12.15	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	87.99	11.58	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	88.95	10.62	0.43	100.00 (61)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	88.95	10.62	0.43	100.00 (61)	0.00 (0)

Table 41: Performance of the ALD 3.0 on Clip 2 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	90.94	8.84	0.21	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	90.73	8.92	0.36	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	91.30	8.27	0.43	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	91.30	8.27	0.43	0.00 (0)	0.00 (0)

Table 42: Performance of the ALD 3.0 on Clip 3 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	87.97	11.42	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	87.97	11.42	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	87.90	11.49	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	90.86	8.53	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	90.86	8.53	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	90.86	8.53	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	90.86	8.53	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	90.94	8.45	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	90.94	8.45	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	90.94	8.45	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	90.94	8.45	0.61	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	90.79	8.60	0.61	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	92.38	6.93	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	92.38	6.93	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	92.53	6.78	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	92.53	6.78	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	92.53	6.78	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	92.53	6.78	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	92.46	6.86	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	92.46	6.86	0.69	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	92.46	6.86	0.69	0.00 (0)	0.00 (0)

Table 43: Performance of the ALD 3.0 on Clip 4 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	97.32	2.59	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	97.78	2.104	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	98.52	1.39	0.09	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	98.52	1.39	0.09	0.00 (0)	0.00 (0)

Table 44: Performance of the ALD 3.0 on Clip 5 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	97.92	1.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	98.11	1.70	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	96.92	2.89	0.19	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	96.92	2.89	0.19	0.00 (0)	0.00 (0)

Table 45: Performance of the ALD 3.0 on Clip 6 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	90.48	9.18	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	90.48	9.18	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	90.48	9.18	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	90.48	9.18	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	90.36	9.30	0.34	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	90.48	9.18	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	92.18	7.48	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	92.86	6.80	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	92.86	6.80	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	92.86	6.80	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	92.86	6.80	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	92.97	6.69	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	92.97	6.69	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	92.86	6.80	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	92.86	6.80	0.34	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	92.97	6.69	0.34	0.00 (0)	0.00 (0)

Table 46: Performance of the ALD 3.0 on Clip 7 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	99.56	0.44	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	99.65	0.35	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	99.74	0.26	0.00	0.00 (0)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	99.74	0.26	0.00	0.00 (0)	0.00 (0)

Table 47: Performance of the ALD 3.0 on Clip 8 for different values of β, $\Delta\theta_{max}$, and $\Delta\rho_{max}$.

Parameters	CO %	MA %	MD %	FD %	TR %
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	70.75	29.25	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	70.75	29.25	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	70.75	29.25	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	70.75	29.25	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	70.67	29.33	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	70.67	29.33	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	70.59	29.41	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	70.59	29.41	0.00	100.00 (65)	0.00 (0)
$\beta = 0.78$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	70.59	29.41	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	80.57	19.43	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	80.57	19.43	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	80.65	19.35	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	80.65	19.35	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	80.42	19.58	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	80.42	19.58	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	80.42	19.58	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	80.42	19.58	0.00	100.00 (65)	0.00 (0)
$\beta = 0.88$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	80.42	19.58	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 104$	90.15	9.85	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 104$	90.15	9.85	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 104$	90.07	9.93	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 120$	90.07	9.93	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 120$	89.99	10.01	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 120$	89.99	10.01	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 3$, $\Delta\rho_{max} = 136$	89.99	10.01	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 5$, $\Delta\rho_{max} = 136$	89.99	10.01	0.00	100.00 (65)	0.00 (0)
$\beta = 0.98$, $\Delta\theta_{max} = 7$, $\Delta\rho_{max} = 136$	89.99	10.01	0.00	100.00 (65)	0.00 (0)

APPENDIX D

ADDITIONAL RESULTS OF THE MCLDW SYSTEM

A histogram of $\phi(n)$ for each clip is provided below. Furthermore, the histograms are categorized into different sections that are based on the mode of operation of the MCLDW system.

D.1 Default

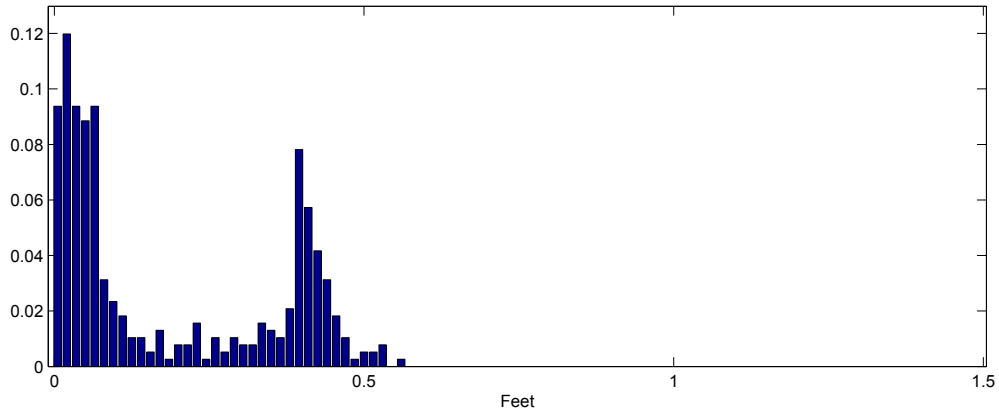


Figure 222: Histogram of $\phi(n)$ for Clip 1. $\sigma_{\phi(n)} = 0.17$

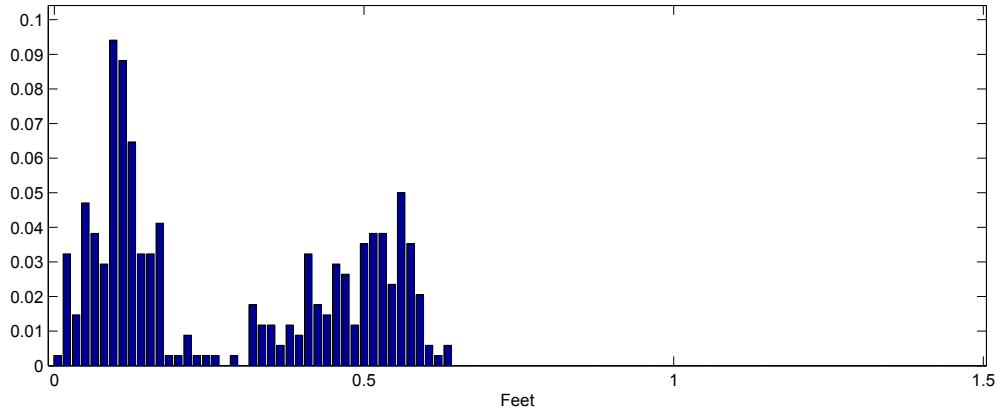


Figure 223: Histogram of $\phi(n)$ for Clip 2. $\sigma_{\phi(n)} = 0.20$

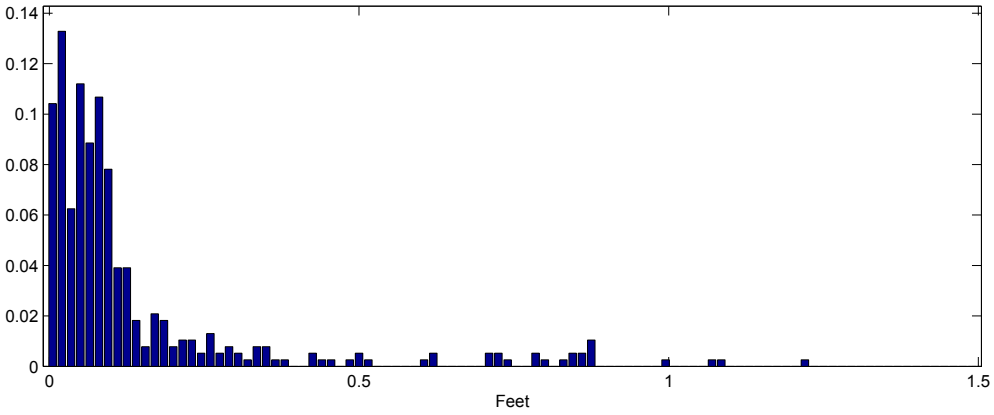


Figure 224: Histogram of $\phi(n)$ for Clip 3. $\sigma_{\phi(n)} = 0.20$

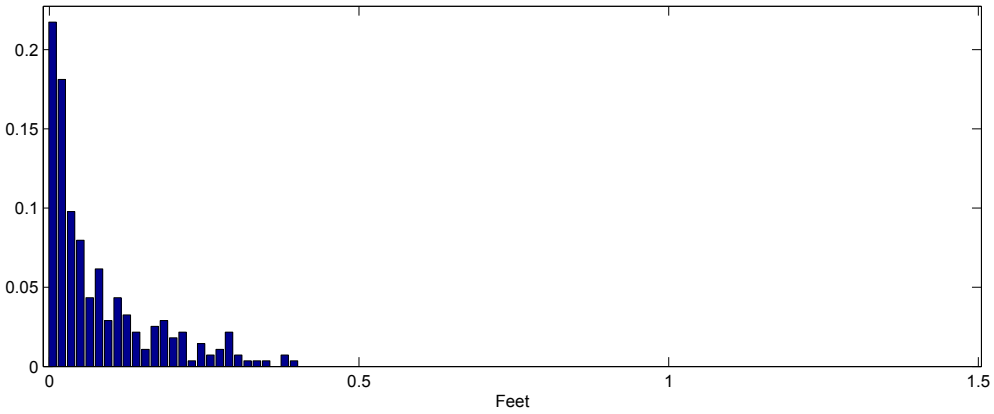


Figure 225: Histogram of $\phi(n)$ for Clip 4. $\sigma_{\phi(n)} = 0.09$

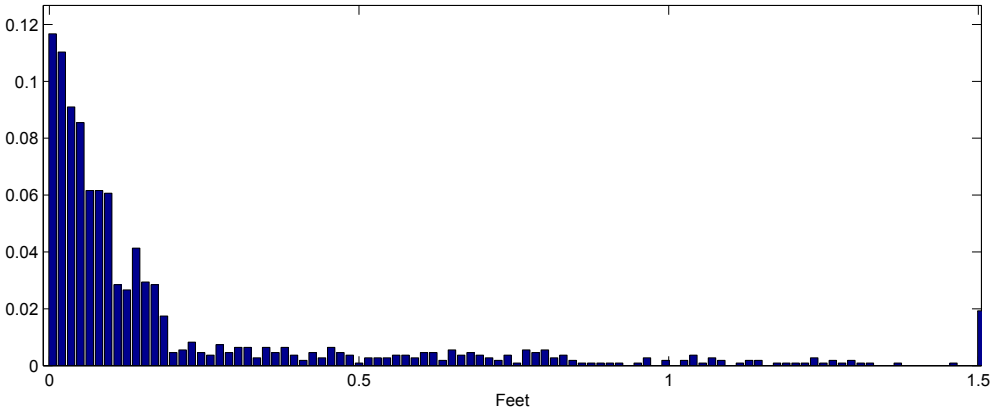


Figure 226: Histogram of $\phi(n)$ for Clip 5. $\sigma_{\phi(n)} = 0.70$

D.2 Spline Replication

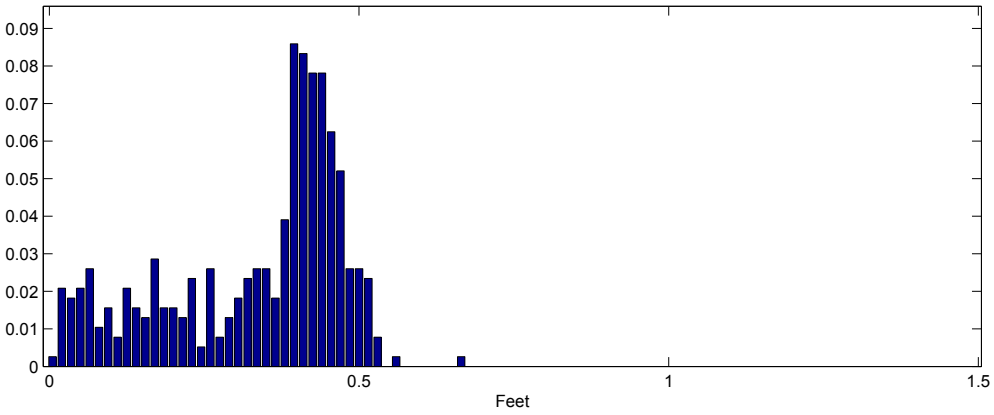


Figure 227: Histogram of $\phi(n)$ for Clip 1. $\sigma_{\phi(n)} = 0.14$

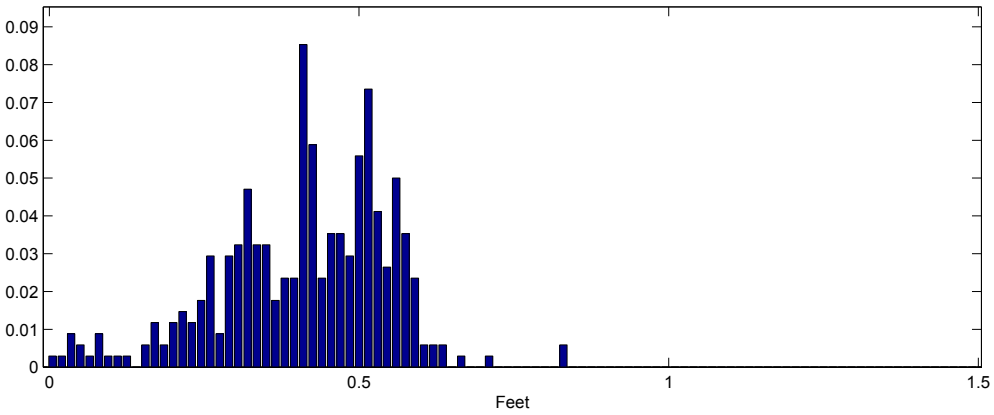


Figure 228: Histogram of $\phi(n)$ for Clip 2. $\sigma_{\phi(n)} = 0.14$

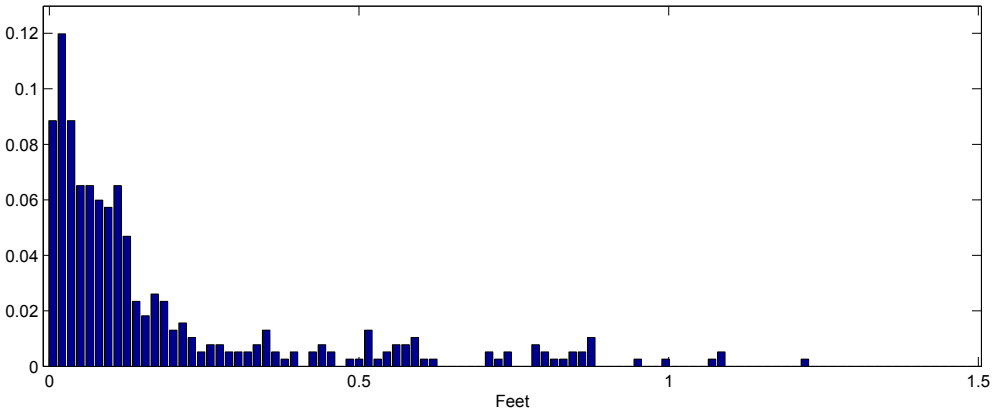


Figure 229: Histogram of $\phi(n)$ for Clip 3. $\sigma_{\phi(n)} = 0.23$

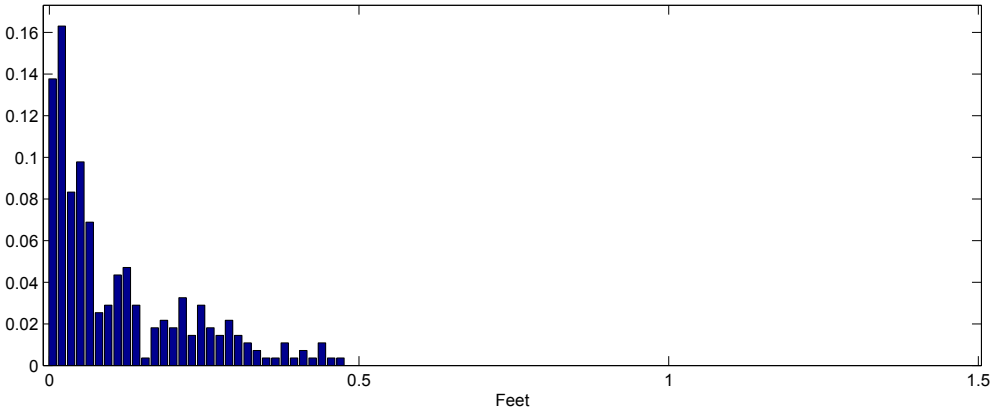


Figure 230: Histogram of $\phi(n)$ for Clip 4. $\sigma_{\phi(n)} = 0.11$

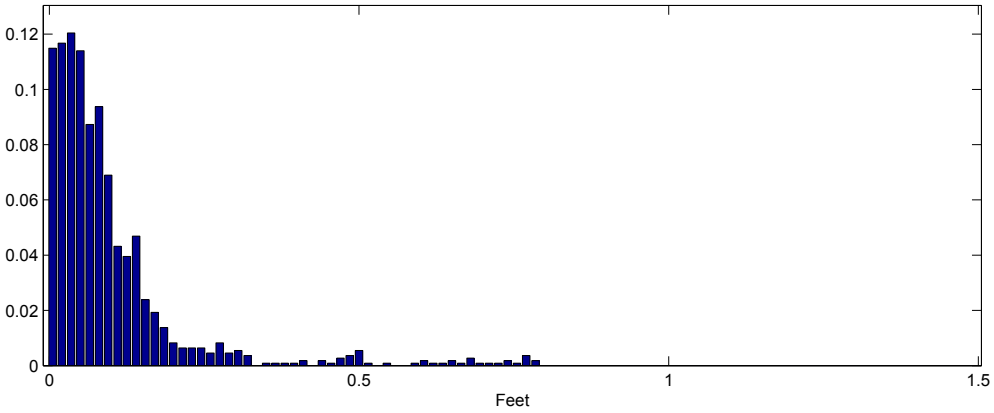


Figure 231: Histogram of $\phi(n)$ for Clip 5. $\sigma_{\phi(n)} = 0.12$

D.3 Boundary Extension

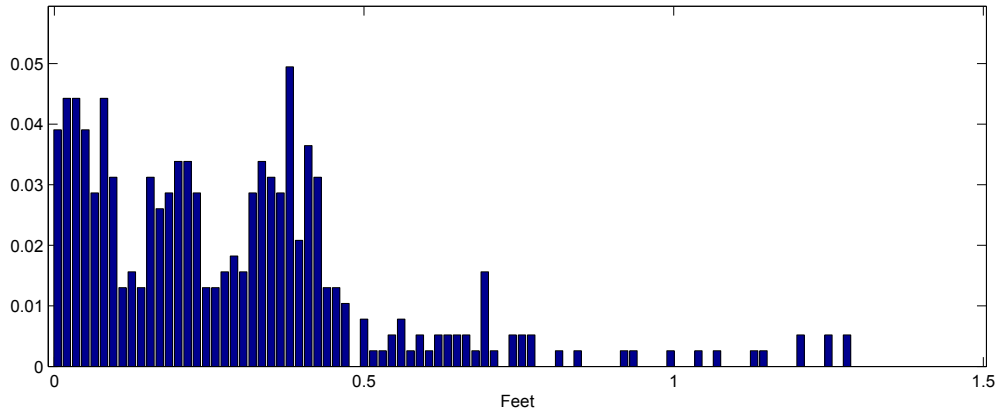


Figure 232: Histogram of $\phi(n)$ for Clip 1. $\sigma_{\phi(n)} = 0.25$

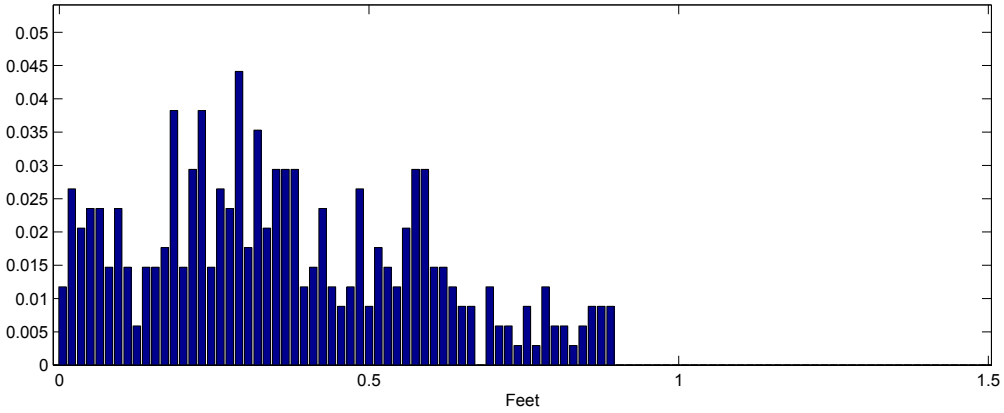


Figure 233: Histogram of $\phi(n)$ for Clip 2. $\sigma_{\phi(n)} = 0.28$

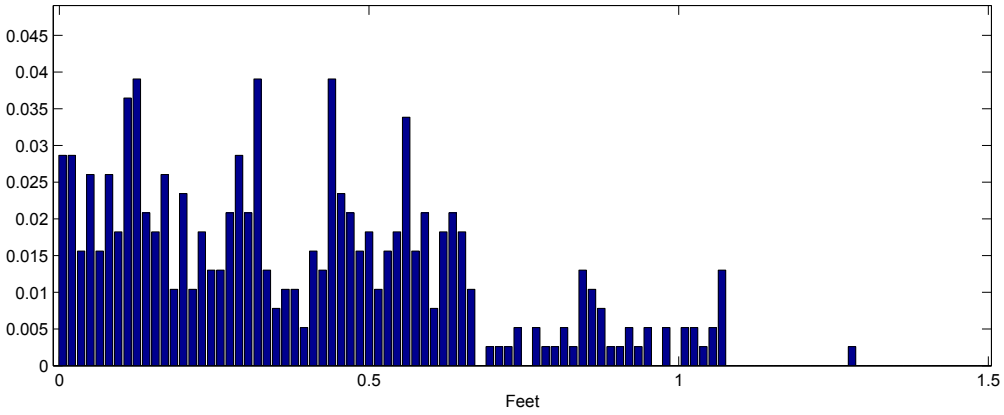


Figure 234: Histogram of $\phi(n)$ for Clip 3. $\sigma_{\phi(n)} = 0.27$

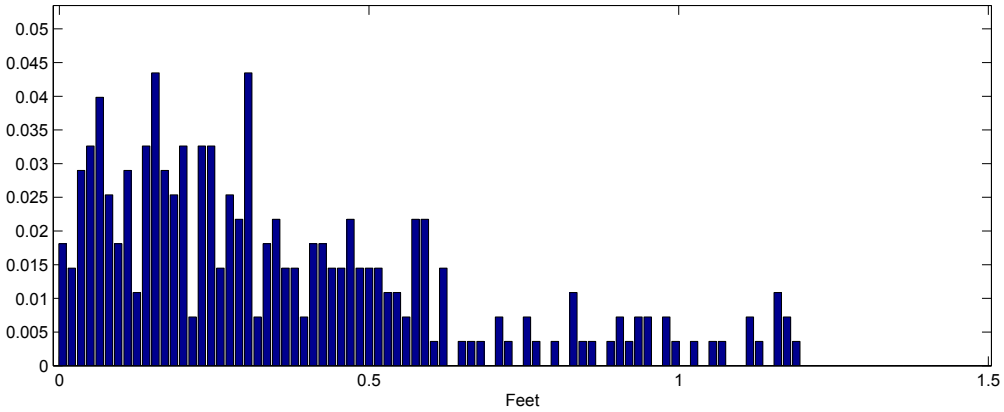


Figure 235: Histogram of $\phi(n)$ for Clip 4. $\sigma_{\phi(n)} = 0.29$

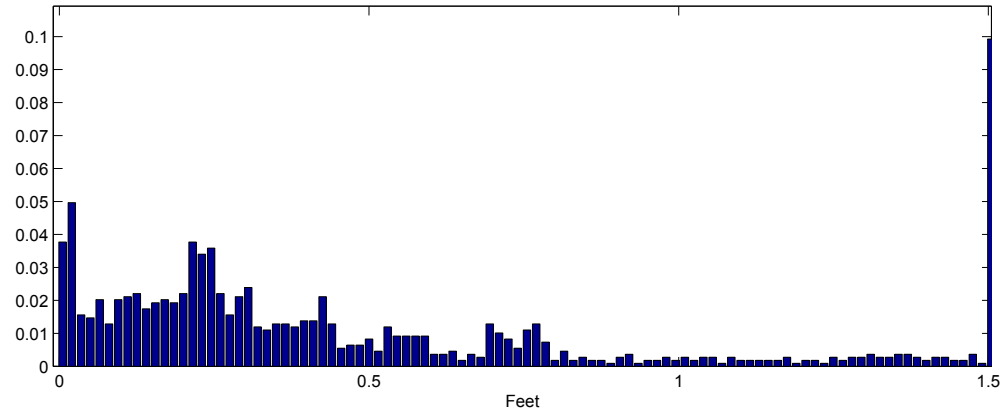


Figure 236: Histogram of $\phi(n)$ for Clip 5. $\sigma_{\phi(n)} = 0.59$

REFERENCES

[1] Bayerische Motoren Werke AG (BMW), "Adaptive cruise control," *BMW Technology Guide*, 2011. [Online]. Available: http://www.bmw.com/com/en/insights/technology/technology_guide/articles/active_cruise_control.html

[2] European New Car Assessment Programme, "Reward 2010 - opel eye," *Euro NCAP Advanced*, 2010. [Online]. Available: http://www.euroncap.com/rewards/opel_eye.aspx

[3] M. Aly, "Real time detection of lane markers in urban streets," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 7–12, 2008.

[4] C. Paukert, "REPORT: NHTSA to mandate lane departure warning and auto-brake systems?" *Autoblog*, 2009. [Online]. Available: http://www.autoblog.com/2009/07/03/report-nhtsa-to-mandate-lane-departure-warning-and-auto-brake-s/

[5] B. Lamprecht, S. Rass, S. Fuchs, and K. Kyamakya, "Extrinsic camera calibration for an on-board two-camera system without overlapping field of view," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 265–270, 2007.

[6] C. Jung and C. Kelber, "A lane departure warning system using lateral offset with uncalibrated camera," in *IEEE Intelligent Transportation Systems*, 2005, pp. 348–353.

[7] "Pixim: Digital Pixel System Technology," 2011. [Online]. Available: http://www.pixim.com/products-and-technology/technology

[8] N. Zheng, S. Tang, H. Cheng, Q. Li, G. Lai, and F. Wang, "Toward intelligent driver-assistance and safety warning system," *IEEE Intelligent systems*, vol. 19, no. 2, pp. 8–11, 2004.

[9] World Health Organization, "Global status report on road safety 2009," *Violence and Injury Prevention and Disability (VIP)*, 2011. [Online]. Available: http://www.who.int/violence_injury_prevention/en/

[10] K. D. Kochanek, J. Xu, S. L. Murphy, A. M. Miniño, and H. C. Kung, "Deaths: Preliminary Data for 2009," *National Vital Statistics Reports*, vol. 59, no. 4, pp. 1–51, 2011.

[11] National Highway Traffic Safety Administration, "FARS Encyclopedia," *Fatality Analysis Reporting System (FARS)*. [Online]. Available: http://www-fars.nhtsa.dot.gov/

[12] G. Jacobs, A. Aeron-Thomas, and A. Astrop, *Estimating global road fatalities*. Transport Research Laboratory, 2000.

[13] Mobileye, "Mobileye C2-170," 2011. [Online]. Available: http://mobileye.com/

[14] Iteris, "Iteris Vehicle Sensors: Lane Departure Warning," 2007. [Online]. Available: http://www.iteris.com/

[15] A. Borkar, M. Hayes, and M. Smith, "An efficient method to generate ground truth for evaluating lane detection systems," *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 1090–1093, 2010.

[16] Bayerische Motoren Werke AG (BMW), "Lane departure warning," *BMW Technology Guide*, 2011. [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2010/safety/drive_assistance/lane_departure_warning_information.html

[17] ——, "Lane change warning," *BMW Technology Guide*, 2011. [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2010/safety/drive_assistance/lane_change_warning.html

[18] Mobileye, "Mobileye binary headlamp contol," *Mobileye Applications*, 2011. [Online]. Available: http://mobileye.com/technology/applications/head-lamp-control/binary-headlamp-contol/

[19] Federal Highway Administration, "Manual Uniform Traffic Control Devices," http://mutcd.fhwa.dot.gov/, Nov. 2009.

[20] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.

[21] M. Aly, "Real time lane detection in urban streets," http://www.vision.caltech.edu/malaa/research/lane-detection/, 2011.

[22] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] A. Borkar, M. Hayes, and M. T. Smith, "Lane Detection And Tracking Using A Layered Approach," *Proceedings of the Advanced Concepts for Intelligent Vision Systems*, pp. 474–484, 2009.

[25] G. Hoffmann, "Luminance Models for Grayscale Conversions," pp. 1–11, 2002.

[26] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and Kalman filter," *Proceedings of the IEEE International Conference on Image Processing*, pp. 3261–3264, 2009.

[27] E. Johnson and R. Hamburger, "Inverse Perspective Mapping: CS5320/6320 Computer Vision Project," College of Engineering, University of Utah, Tech. Rep., 2007.

[28] E. R. Davies, *Machine Vision : Theory, Algorithms, Practicalities*, 3rd ed. Morgan Kaufmann, 2004.

[29] A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti, "A Layered Approach To Robust Lane Detection At Night," *Proceedings of the IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, pp. 51–57, 2009.

[30] J. P. Lewis, "Fast normalized cross-correlation," *Proceedings of the CIPPRS Vision Interface*, pp. 120–123, 1995.

[31] A. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[32] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall, 2002.

[33] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons, 1996.

[34] D. Simon, "Kalman filtering," *Embedded Systems Programming*, vol. 14, no. 6, pp. 72–79, 2001.

[35] A. Borkar, M. Hayes, and M. T. Smith, "Detecting Lane Markers In Complex Urban Environments," *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 727–732, 2010.

[36] Intel Corporation, "Intel Core 2 Quad Processors," 2011. [Online]. Available: http://www.intel.com/products/processor/core2quad/

[37] Microsoft, "Windows XP," 2011. [Online]. Available: http://windows.microsoft.com/en-US/windows/products/windows-xp

[38] International Telecommunication Union, "Rec. ITU-R BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios," 2011.

[39] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[40] A. Borkar, M. Hayes, and M. T. Smith, "A Template Matching and Ellipse Modeling Approach to Detecting Lane Markers," *Proceedings of the Advanced Concepts for Intelligent Vision Systems*, pp. 179–190, 2010.

[41] ——, "Polar randomized hough transform for lane detection using loose constraints of parallel lines," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1037–1040, 2011.

[42] ——, "Lane Detection Using Constraints Of Parallel Lines," *Proceedings of the IEEE International Conference on Consumer Electronics*, 2011.

[43] L. Xu, E. Oja, and P. Kultanen, "Randomized hough transform," *Proceedings of the International Conference on Pattern Recognition*, pp. 631–635, 1989.

[44] J. Wang, Y. Wu, Z. Liang, and Y. Xi, "Lane Detection Based on Random Hough Transform on Region of Interesting," *Proceedings of the IEEE International Conference on Information and Automation*, pp. 1735–1740, 2010.

[45] C. Hari, J. Jojish, S. Gopi, V. Felix, and J. Amudha, "Mid-Point Hough Transform: A Fast Line Detection Method," *Proceedings of the IEEE India Conference*, pp. 1–4, 2009.

[46] Infiniti USA, "2012 Infiniti M Specs & Options," 2011. [Online]. Available: http://www.infinitiusa.com/m/specs-options

[47] ROS.org, "Laser Camera Calibration," 2009. [Online]. Available: http://www.ros.org/wiki/laser_camera_calibration

[48] Center for Advanced Spatial Technologies, "Camera Calibration with PhotoStruct," 2011. [Online]. Available: http://gmv.cast.uark.edu/2285/camera-calibration-with-photostruct-2/

[49] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 187–194, 2004.

[50] K. Shafique, A. Hakeem, O. Javed, and N. Haering, "Self Calibrating Visual Sensor Networks," *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 1–6, 2008.

[51] K. Tieu, G. Dalley, and W. Grimson, "Inference of non-overlapping camera network topology by measuring statistical dependence," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1842–1849 Vol. 2, 2005.

[52] Q. Wang, Y. Liu, and Y. Shen, "An accurate extrinsic camera self-calibration method in non-overlapping camera sensor networks," *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, pp. 1–6, 2011.

[53] F. Pagel, "Calibration of Non-Overlapping Cameras in Vehicles," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1178–1183, 2010.

[54] R. Kumar, A. Ilie, J. Frahm, and M. Pollefeys, "Simple Calibration of Non-overlapping Cameras with a Mirror Department of Computer Science," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, 2008.

[55] P. Lébraly, C. Deymier, O. Ait-Aider, E. Royer, and M. Dhome, "Flexible Extrinsic Calibration of Non-Overlapping Cameras Using a Planar Mirror : Application to Vision-Based Robotics," *IEEE Intelligent Robots and Systems*, pp. 5640–5647, 2010.

[56] Point Grey, "Ladybug2 (1394b)," *Spherical Products*, 2011. [Online]. Available: http://www.ptgrey.com/products/ladybug2/ladybug2_360_video_camera.asp

[57] A. Borkar, M. Hayes, and M. T. Smith, "A non overlapping camera network: Calibration and application towards lane departure warning," *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2011.

[58] H. Lin, S. Ko, Y. Kim, and H. Kim, "Lane departure identification on highway with searching the region of interest on hough space," *Proceedings of the International Conference on Control, Automation and Systems*, pp. 1088–1091, 2007.

[59] Y. Leng and C. Chen, "Vision-Based Lane Departure Detection System in Urban Traffic Scenes," *Proceedings of the International Conference on Control Automation Robotics & Vision*, pp. 7–10, 2010.

[60] C. Kelber, "A lane departure warning system based on a linear-parabolic lane model," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 891–895.

[61] J. G. Wang, C. J. Lin, and S. M. Chen, "Applying fuzzy method to vision-based lane detection and departure warning system," *Expert Systems with Applications*, vol. 37, no. 1, pp. 113–126, 2010.

[62] J. Lee, "A Machine Vision System for Lane-Departure Detection," *Computer Vision and Image Understanding*, vol. 86, no. 1, pp. 52–78, 2002.

[63] J. W. Lee and U. K. Yi, "A lane-departure identification based on LBPE, Hough transform, and linear regression," *Computer Vision and Image Understanding*, vol. 99, no. 3, pp. 359–383, 2005.

[64] X. An, M. Wu, and H. He, "A Novel Approach to Provide Lane Departure Warning Using Only One Forward-Looking Camera," *Proceedings of the International Symposium on Collaborative Technologies and Systems*, pp. 356–362, 2006.

[65] G. Gerber, R. Desonia, R. D. Ervin, C. F. Lin, A. G. Ulsoy, and T. E. Pilutti, "CAPC: An Implementation of a Road Departure Warning System," *Proceedings of the IEEE International Conference on Control Applications*, pp. 590–595, 1996.

[66] X. Dai, A. Kummert, S. B. Park, and D. Neisius, "A warning algorithm for Lane Departure Warning system," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 431–435, 2009.

[67] D. J. Leblanc, G. E. Johnson, P. J. T. Venhovens, G. Gerber, R. Desonia, R. D. Ervin, C. F. Lin, A. G. Ulsoy, and T. E. Pilutti, "CAPC: A Road-Departure Prevention System," *IEEE Control Systems*, vol. 16, no. 6, pp. 61–71, 1996.

[68] W. Mo, X. An, and H. He, "On Vision-based Lane Departure Detection Approach," *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, pp. 353–357, 2005.

[69] F. Cosşkun, O. Tunçer, M. Karsligil, and L. Güvenç, "Real Time Lane Detection and Tracking System Evaluated in a Hardware-in-The-Loop Simulator," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 1336–1343, 2010.

[70] H. Kim, S. Hong, H. Son, and F. Werblin, "High Speed Road Boundary Detection On The Images," *Proceedings of the International Symposium on Circuits and Systems*, no. 5, pp. 769–772, 2003.

[71] W. Kwon, J. W. Lee, D. Shin, K. Roh, D. Y. Kim, and S. Lee, "Experiments on decision making strategies for a lane departure warning system," *Proceedings of the IEEE International Conference on Robotics and Automation*, no. May, pp. 2596–2601, 1999.

[72] S. Ieng, J. Vrignon, D. Gruyer, and D. Aubert, "A New Multi-Lanes Detection Using Multi-Camera for Robust Vehicle Location," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 700–705, 2005.

[73] E. Catmull and R. Rom, "A class of local interpolating splines," *Computer Aided Geometric Design*, vol. 74, pp. 317–326, 1974.

[74] American Association of State Highway and Transportation Official (AASHTO), "Recommendations for AASHTO Superelevation Design," 2003. [Online]. Available: http://downloads.transportation.org/SuperelevationDiscussion9-03.pdf

[75] Mercedes-Benz USA, "2011 Mercedes-Benz S550 Specs," 2011. [Online]. Available: http://www.mbusa.com/mercedes/vehicles/explore/specs/class-S/model-S550V

[76] "Extensible Markup Language (XML) 1.0 (Fifth Edition)," 2008. [Online]. Available: http://www.w3.org/standards/xml/

[77] R. Frischholz, "Face Detection Datasets," 2011. [Online]. Available: http://www.facedetection.com/facedetection/datasets.htm

[78] Center for Machine Learning and Intelligent Systems, "Letter Recognition Data Set," 2011. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Letter+Recognition

[79] T. Veit, J. P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of Road Marking Feature Extraction," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 174–181, 2008.

[80] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D Scene Analysis from a Moving Vehicle," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[81] C. C. Wang, "CMU/VASC image database," 2003. [Online]. Available: http://vasc.ri.cmu.edu/idb/html/road/index.html

[82] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and Recognition using Structure from Motion Point Clouds," *Proceedings of the European Conference on Computer Vision*, pp. 1–14, 2008.

[83] D. Makris, "PETS2001 Dataset," 2001. [Online]. Available: http://www.cvg.cs.rdg.ac.uk/PETS2001/pets2001-dataset.html

[84] S. Sivaraman and M. M. Trivedi, "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, Jun. 2010.

[85] V. Santos, J. Almeida, D. Gameiro, M. Oliveira, R. Pascoal, R. Sabino, and P. Stein, "ATLASCAR Technologies for a Computer Assisted Driving System on board a Common Automobile," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 1421–1427, 2010.

[86] K. H. Lim, A. Cat, L. E. Ngo, K. P. Seng, and L.-M. Ang, "UNMC-VIER AutoVision Database," *Proceedings of the Conference on Computer Applications and Industrial Electronics*, pp. 650–654, 2010.

[87] Multimedia Imaging Technology, "Image Sequence Analysis Test Site (EISATS)." [Online]. Available: http://www.mi.auckland.ac.nz/EISATS/

[88] E. Reinhard, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2006.

[89] "The Imaging Source: Frame Grabbers," 2011. [Online]. Available: http://www.theimagingsource.com/en_US/products/grabbers/dfgmc4pcie/

[90] "OBD Pros Scantool," 2011. [Online]. Available: http://www.obdpros.com/

[91] "SAE International: On-Board Diagnostic Standards," 2011. [Online]. Available: http://standards.sae.org/electrical-electronics-avionics/obd/standards/

[92] "ISO/IEC 15948:2004–Information technology–Computer graphics and image processing–Portable Network Graphics (PNG)," 2004.

[93] "WinRAR," 2011. [Online]. Available: http://www.win-rar.com/

[94] D. Samardzija, E. Kovac, D. Isailovic, B. Miladinovic, N. Teslic, and M. Katona, "Road Nail : Intelligent Road Marking System Testbed," *Proceedings of the IEEE Vehicular Networking Conference*, pp. 134–138, 2010.

[95] ——, "Road Nail : Intelligent Road Marking System," *Proceedings of the IEEE International Conference on Consumer Electronics*, pp. 571–572, 2011.

VITA

Amol Borkar was born in Mumbai, India in 1982. He received his B.S. in Computer Engineering from North Carolina State University and his M.S. in Electrical Engineering from the Georgia Institute of Technology in 2004 and 2007, respectively. He is currently a Ph.D. student at the Center for Signal and Image Processing at Georgia Tech under the co-advisement of Dr. Monson H. Hayes and Dr. Mark T. Smith (KTH - Swedish Royal Institute of Technology). He is expected to receive his Ph.D. degree in 2011. During his academic tenure, he has gained extensive industry experience through multiple internships. During the course of his bachelor's degree, he was employed by P.B.J. industries in Mumbai, India, in 2001, and by Intel Corporation, Beaverton, Oregon, in 2003. During his graduate program, he spent 8 months at IBM's Watson Research Center, New York in 2008. His primary research areas are applied computer vision, image processing, pattern recognition, and machine learning. He also has multi-disciplinary research experience in the fields of automotive vision, face detection, face recognition, and medical imaging.